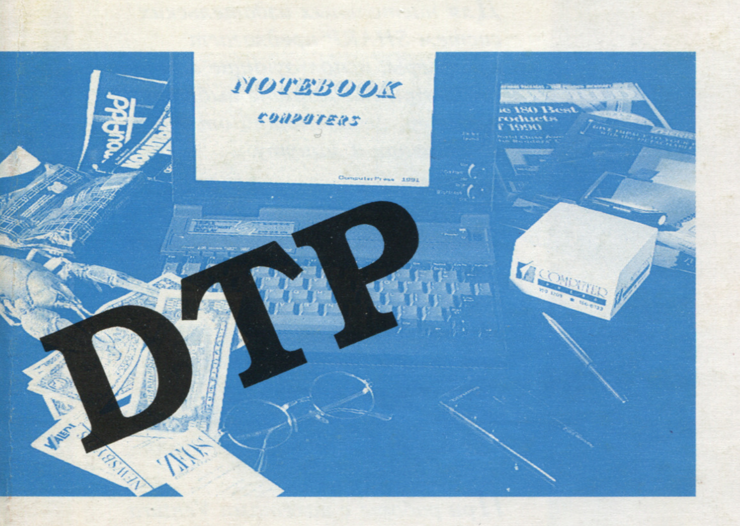
ISSN 0868-6157

КОМПЬЮТЕР ПРЕСС

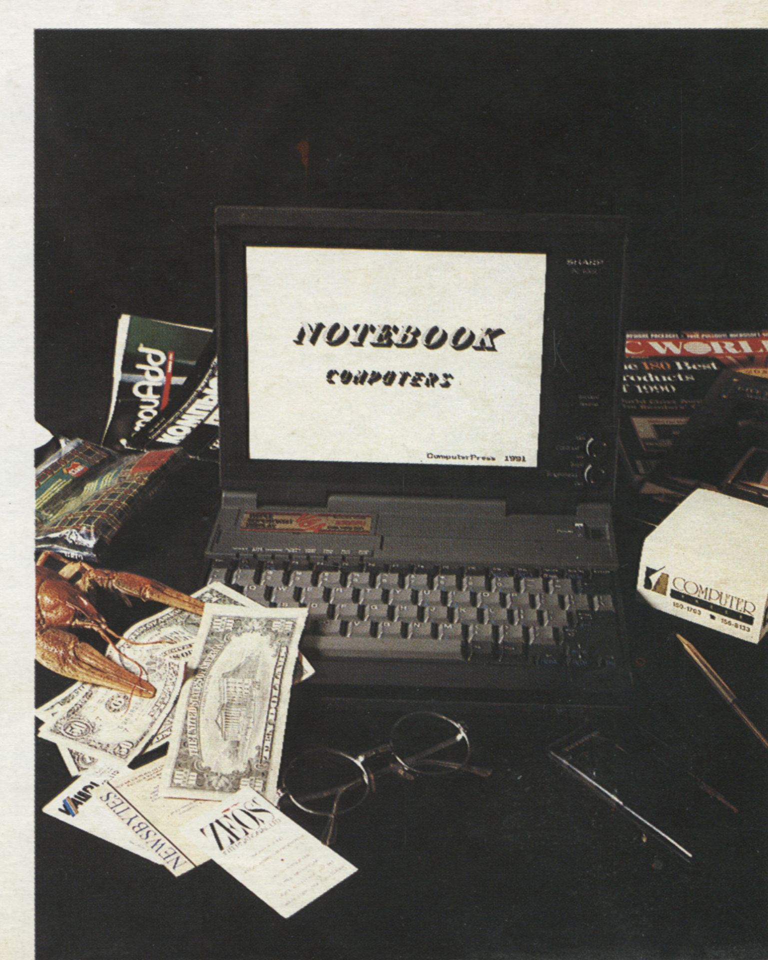








3'92



Совместное советско-американское предприятие «СОВАМИНКО»

КОМПЬЮТЕР ПРЕСС

ИЗДАТЕЛЬСКИЕ СИСТЕМЫ

Настольные издательства от А до Я 5

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

Архитектура микропроцессора i80486 15

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Microsoft Visual BASIC: первые шаги в программировании для Windows 21

Quattro Pro 3.0: все, что нужно для продуктивной работы 33

БЕЗОПАСНОСТЬ КОМПЬЮТЕРНЫХ СИСТЕМ

Программные и аппаратные средства защиты информации для персональных компьютеров 37

Американская классификация и принципы оценивания безопасности компьютерных систем 47

РАБОТАЕМ ГРАМОТНО

Графические форматы данных 55

БАЗЫ ДАННЫХ

dBASE IV для начинающих 59

СЕТИ

Каталог продуктов фирмы Novell 65

ТЕНДЕНЦИИ

Современный потребитель: эволюция потребностей 75

НОВОСТИ 77

3'92

Итоги конкурса на лучшую статью 1991 года

Итак, дражайшие читатели, давайте подведем некоторые итоги. Вначале о тех, чьи творения мы читали. Их можно разделить на три группы. К первой можно отнести тех, к кому в полной мере относятся следующие слова:

Хвалю я пьесу Вашу, сэр,
Особенно вторую часть.
Но почему бы, например,
Вам и начало не украсть?

К счастью, эта группа оказалась не очень велика, и в нее, в основном, попали те, кто, будем надеяться, по незнанию, слали нам переводы из западных журналов, чего делать явно не стоило.

Вторая группа была более многочисленна. К ней относятся люди типа изобретателя из Ханты-Мансийска, сумевшего, по его заверению, спаять лучший в мире аналоговый вход для подключения к персональному компьютеру электрогитары, о чем подробно рассказывается в небольшой семнадцатистраничной статье. Таких людей, как, впрочем, и организаций, оказалось крайне немало, а в своих статьях они рекламировали порой столь необычные вещи, что мы всей редакцией рылись по справочникам, пытаясь понять, о чем идет речь. Например, запомнился автор, приславший многотомное описание разработанного им пакета для "перепрограммирования универсального мультиколлектора".

Ну, и к третьей группе относятся пытавшиеся что-то написать, но не сумевшие. Иногда причиной этого являлось отсутствие знаний о предмете изложения, иногда незнание русского языка, причем, незнание в такой степени, что при всем старании редакции статью не удавалось привести в божеский вид.

Плодовитый поэт из Канады
Безразмерные пишет баллады.
Рецензенты ворчат:
"Этот стих длинноват",
А он отвечает:"Я люблю последнюю
строчку делать намного длиннее, чем надо".

Теперь о тех, чьи статьи читали и мы, и вы. То есть о том, что мы печатали. Таких, как вы понимаете, за год набралось изрядно, и неудивительно, что при обсуждении итогов редакция столкнулась с некоторыми трудностями. Эту ситуацию точно изложил Льюис Кэрролл в "Охоте на Снарка":

Впрочем, мненья присяжных сложились давно,
Всяк отстаивал собственный взгляд,
И решительно было ему все равно,
Что коллеги его говорят.

Тем не менее, итоги подвести удалось, и мы с гордостью называем победителей конкурса 1991 года.

1 премия:
И.Моисеенков.
За статьи:
Суета вокруг Роберта или Моррис-сын и все-все-все. №8, 9.
Основы безопасности компьютерных систем. №10, 11, 12.

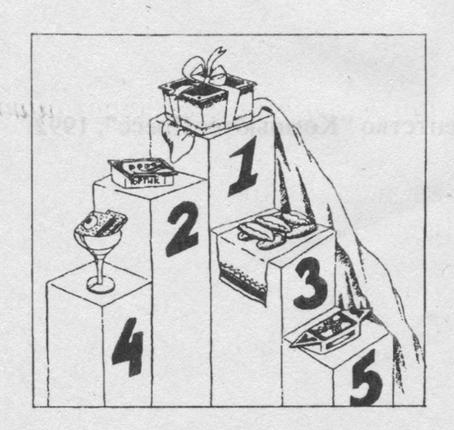
2 премия:
Е.Касперский.
За статьи:
Компьютерные вирусы: предварительные соображения. №5, 6, 7.
Дыры в MS-DOS и программы защиты информации. №10.

3 премия:
А.Петроченков.
За статьи:
Переселение в глобальную деревню. №12.
10 лет IBM PC. №12.
PC Tools 7.0. Интегрированный профессиональный инструментарий. №11.

Поощрительные премии:

1. **П.Антонова.** Сеть Релком и электронная почта. №10.
2. **Г.Берг.** Локальные сети: а стоит ли из пушки по воробьям? №3.
3. **А.Гиглавый, К.Чащин.** Двадцать игроков решают пойти с туза. №5. Новейшая история компьютерных войн: кошмары по Оруэллу. №9.
4. **Н.Зайцев.** FoxPro! Боевик из жизни программ. №4.
5. **А.Колесов.** QuickBasic — это то, что Вам нужно. №3.
6. **Ю.Кравацкий.** Русский драйвер экрана и клавиатуры. №10.
7. **А.Матвеев.** От С к С++. Записки хакера. №9.
8. **С.Новосельцев.** Мультимедиа — синтез трех стихий. №7, 8, 11.
9. **И.Свиридов.** Введение в телекоммуникации. №3.
10. **А.Федоров.** За серию статей о продуктах фирмы Borland. №11, 12.

Мы поздравляем наших победителей с достойным выступлением, в заключение добавив, что, поскольку рубль стал несколько не тем, каким был на момент объявления конкурса, редакция решила увеличить размер премий в три раза. С чем мы также поздравляем наших лауреатов и ждем от них (и не только от них) новых услаждающих душу и радующих разум творений.



Б.Молчанов
Главный редактор



КОМПЬЮТЕРПРЕСС

Издается с 1989 года
Выходит 12 раз в год
3'92 (27)

В номере использована графика
М.К.Эшера

Адрес редакции:
113093 Москва, аб.ящик 37
Факс: (095) 200-22-89
Телефоны для справок:
(095) 491-01-53, 420-83-80
E-mail: postmaster@Computerpress.msk.su

Сдано в набор 2.02.92. Подписано к печати 12.02.92.
Формат 84x108/16. Печать офсетная. Бумага офсетная.
Усл.печ.листов 8,4+0,42 (обл.). Тираж 100000 экз.
(1-й завод — 62000). Заказ 2578. С-3.

Оригинал-макет подготовлен агентством
«КомпьютерПресс»

Отпечатано в полиграфической фирме «Красный
пролетарий» РГИИЦ «Республика».
103473 Москва, И-473, Краснопролетарская, 16.



Настольные издательства от А до Я

Как делается журнал, книга или газета? Вообще-то, по-разному. Например, так.

Кто-то — автор — пишет текст. Художественный редактор разрабатывает внешнее и внутреннее оформление издания, называемое макетом. Этот человек определяет формат издания, количество колонок текста на странице, расстояние между колонками и абзацами, отступы от краев страницы — т.н. поля, размер и вид используемых шрифтов, местонахождение и размер иллюстраций и много других важных вещей.

Этот макет передается верстальщику — человеку, который должен постранично размещать передаваемый ему материал (текст, иллюстрации, таблицы и пр.) в соответствии с этим макетом и полиграфическими требованиями.

До передачи верстальщику авторский текст проверяется литературным редактором, который устраняет допущенные автором литературные погрешности, и научным редактором, отвечающим за смысловую часть и терминологию.

П сле этого текст, если он был передан в рукописном виде, вводится наборщиком в компьютер и в виде файла передается верстальщику.

Затем текст проверяется корректором — человеком или программой (а лучше и то, и другое) — для устранения орфографических ошибок и опечаток.

После подготовки верстальщиком материал еще раз проверяется корректором — не было ли внесено случайных ошибок в процессе верстки — и передается в распечатанном виде техническому редактору. Последний проверяет, соответствует ли распечатанный материал макету художественного редактора, не допущены ли ошибки при верстке, и готовит спецификацию на макет (проще говоря, пишет пояснительную записку для типографии, в которой будет печататься это издание).

По окончании работы технического редактора, макет вместе со всеми сопроводительными материалами проверяется главным редактором, подписывается им в печать и отправляется в типографию.

Все, о чем сказано выше, называется издательским процессом. Пакеты программ настольного издательства (desktop publishing, или пакеты DTP), о которых пойдет речь ниже, по сути, являются инструментом верстальщика. Т.е. программа настольного издательства позволяет легко манипулировать текстом, менять форматы страниц, размер отступов, дает возможность комбинировать разные шрифты, короче, работать с материалом до получения полного удовлетворения от внешнего вида как отдельных страниц (полос издания), так и всего издания.

Отличия между издательскими пакетами и текстовыми процессорами

Даже в нашей отсталой стране текстовые процессоры уже получили достаточно широкое распространение. По ряду функциональных возможностей пакеты DTP аналогичны лучшим текстовым

ОСНОВНЫЕ ПОЛИГРАФИЧЕСКИЕ ТЕРМИНЫ

Body text (текстовое поле). Основной текст документа. Термин используется, чтобы было ясно, что речь идет именно о тексте, а не о заголовках, подписях под рисунками и пр.

Bullet (пуля). Черная точка, звездочка или другой символ, указывающий пункт в списке.

Drop Cap (буквица). Большая первая буква, которая имеет существенно больший размер, чем остальные и может находиться ниже уровня первой строки абзаца.

Em, En (эм, эн). Единицы ширины. Em – это ширина используемого шрифта в пунктах. Свое название Em получила из-за того, что ширина стандартной буквы M равна ее высоте. En равна половине Em. Эти единицы часто используются для указания размера специальных символов, например, тире.

Font (шрифт). Строго говоря, это гарнитура определенного кегля и типа, например, Гельветика жирная 12 пунктов. Во многих издательских пакетах используется как синоним понятия "гарнитура".

Gutter (средник). Расстояние между колонками.

Kerning (кернинг). Сдвиг соседних букв слова ближе друг к другу, чтобы ликвидировать слишком большие расстояния между ними.

Justification (выравнивание). Известный термин, применяемый почти во всех более-менее развитых текстовых редакторах, означающий выравнивание программой строк абзаца по обеим границам текстового поля. Если ваш издательский пакет не снабжен программой переноса русских слов, то выравнивание будет достигаться введением больших дополнительных пробелов между словами, в результате чего текст потеряет свою привлекательность.

Leading (интерлиньяж). Расстояние между строками абзаца. Измеряется в тех же типографских пунктах.

Ligature (лигатура). Символ, представляющий собой два слитых воедино символа.

Line Space – см. **Leading**.

Loose lines (жидкие строки). Порой неудачно выбранное расстояние между символами в слове или словами в строке приводит к тому, что строки становятся разорванными. В некоторых пакетах DTP программа может автоматически выделять такие "ущербные" строки.

Tight lines (слившиеся строки). Порой неудачно выбранное расстояние между символами в слове или словами в строке приводит к тому, что символы в слове или слова в строке налезают друг на друга. В некоторых пакетах DTP программа имеет средства автоматической защиты от этой ситуации.

Orphan (сирота). Последняя строка на странице, являющаяся первой строкой нового абзаца или заголовка. По каноническим правилам полиграфии это считается ошибкой верстки.

Widow (вдова). Первая строка на новой странице, являющаяся последней строкой абзаца или заголовка. По правилам полиграфии это считается ошибкой верстки.

Orphan-Widow control (управление висячими строками). Программные средства многих издательских пакетов, позволяющие устранить ошибки верстки, указанные в двух последних пунктах. (Оторвавшиеся от своего абзаца строки по-русски называются висячими строками, независимо от того, расположены они в начале или в конце страницы.)

Reversed out text (выворотка). Текст, "вывернутый наизнанку", например, белый текст на черном фоне.

Rules (линейки). Горизонтальные или вертикальные линии с нанесенной шкалой, служащие для контроля за размером текстового поля. Кроме этого, элементы оформления издания, представляющие собой горизонтальные или вертикальные линии, отделяющие элементы макета друг от друга.

Resolution (разрешение). Все "компьютерные" изображения, как на экране монитора, так и выводимые на принтер, состоят из рядов и колонок отдельных точек. Чем выше разрешение устройства, тем больше точек помещается на единице поверхности экрана или бумаги и тем выше качество получающегося изображения.

Tracking (разрядка или уплотнение). Увеличение или уменьшение расстояний между отдельными символами в словах, обычно для того, чтобы растянуть длину строки до уровня остальных строк колонки.

Typeface (гарнитура). Группа шрифтов разных кеглей и видов, имеющих одинаковое начертание.

процессорам, и граница, разделяющая их, становится все незаметнее. Так, оба типа программ позволяют размещать на одной странице текст и иллюстрации, оба типа позволяют форматировать текст в несколько колонок, оба типа могут быть использованы для редактирования текста и манипулирования текстовыми блоками. Причем, порой в текстовых процессорах тот же эффект достигается значительно более простым и быстрым путем. Это связано с тем, что они являются существенно более простым инструментом, чем пакеты DTP. Например, в журнале КомпьютерПресс ряд страниц сверстан в пакете DTP Ventura Publisher 2.0, а ряд — для ускорения работы — в текстовом процессоре Microsoft Word 5.0.

Но пакеты DTP отличаются от текстовых процессоров еще двумя важными характеристиками. Во-первых, пакеты DTP имеют более широкие возможности управления подготовкой текста. Во-вторых, подготовленные в пакете DTP материалы выглядят изданиями высшего уровня качества, а не просто изящные распечатки. На этих отличиях мы остановимся ниже.

Что такое PostScript?

PostScript (ПостСкрипт) – это язык программирования, являющийся по своей сути языком описания страниц, который "понимают" некоторые лазерные принтеры и другие устройства вывода.

Последовательность постскриптовских команд, переданная на такой принтер, сообщает ему в точности, как должна в итоге выглядеть отпечатанная страница.

Например, существуют специальные команды для вычерчивания прямых, окружностей, заполнения отдельных областей тем или иным видом штриховки, печати текста тем или иным шрифтом. Вся печать осуществляется в тесной привязке к условной сетке координат, которая накладывается на страницу того или иного формата.

Во многих издательских пакетах и текстовых процессорах существует возможность создания таких постскриптовских команд, которые затем могут быть переданы непосредственно на принтер или записаны на диск в виде постскриптовского файла.

Если у вас нет постскриптовского принтера, вы можете отправиться с дискетой, на которой записан постскриптовский файл, к кому-то, являющемуся счастливым обладателем постскриптовского принтера.

Если вы работаете в среде MS Windows, то вам доступна еще одна альтернатива: использование пакета Adobe Type Manager. Он позволяет выводить работы, подготовленные в PostScript-виде, на непостскриптовский принтер.

На каждом постскриптовском принтере есть своя библиотека постскриптовских гарнитур, благодаря чему появляется возможность манипулировать со шрифтами, сжимая и растягивая их до получения шрифта нужного размера. К этой библиотеке могут быть легко добавлены дополнительные гарнитуры.

PostScript довольно умен – при работе с устройством вывода он всегда заставляет последнее работать с максимально возможным разрешением, поэтому один и тот же постскриптовский файл может быть распечатан на одном принтере с разрешением 300 точек на дюйм, а на другом с разрешением 600 точек на дюйм, в зависимости от способностей каждого из них. В этом случае текстовое поле обеих страниц будет иметь одинаковые размеры, но вторая страница будет несоизмеримо качественнее.

Управление подготовкой текста

Если текстовые процессоры ориентированы на работу со словами и абзацами, то пакеты DTP позволяют легко и элегантно манипулировать текстом до уровня отдельных символов в слове. Чтобы облегчить процесс подготовки страницы, все пакеты DTP снабжены возможностью вывода на экран монитора точной копии того, что будет распечатано на принтере, — эта характеристика называется по-английски WYSIWYG (What You See Is What You Get) — что видишь, то получишь. Эта штука позволяет сэкономить тонны бумаги и годы времени, поскольку для того, чтобы заметить и исправить ошибку на странице, не требуется выводить ее на печать.

Справедливости ради надо заметить, что лучшие текстовые процессоры типа Microsoft Word или WordPerfect имеют возможность просмотра на экране готовой страницы, но только всей целиком, из-за чего не видны отдельные детали. А в пакете DTP можно просмотреть в увеличенном виде любой подозрительный участок страницы.

Все пакеты DTP имеют характеристики, отсутствующие в абсолютном большинстве текстовых процессоров, например, сжатие и растяжение строк, вращение текста и изменение расстояний между строчками и абзацами с очень маленьким шагом приращения.

Скажем, в Microsoft Word 5.0 существует возможность изменения расстояния между строчками абзаца, но шаг приращения слишком велик, и часто случается так, что абзац не становится в отведенное ему место, а при уменьшении расстояния между строчками на один шаг, они начинают налезать одна на другую — в результате приходится переверстывать всю страницу.

До появления пакетов DTP оператор, работая в старомодных текстовых процессорах типа распространенного у нас Лексикона, должен был наряду с текстом вводить специальные символы, указывая, какой шрифт он собирается использовать, каким должен быть размер отступа от края страницы, расстояние между строчками и пр. Работающему в пакете DTP все это не нужно, он может не тратить время на поиск в таблице нужного кода-спецсимвола, сосредотачиваясь на творческом процессе.

Распечатка

При выборе программы DTP вы должны заранее решить, как вы будете выводить свои материалы. Существуют два принципиально разных способа: с помощью обычного матричного принтера, используемого, скажем, в вашей бухгалтерии, или с помощью профессионального оборудования.

Что выбрать — это зависит главным образом от того, сколько страниц вы собираетесь печатать в день или час, и от того, какое качество получаемых распечаток вас удовлетворит. Однако, решив использовать уже имеющийся старенький принтер, не забудьте проверить, совместим ли он с выбранным вами пакетом DTP. Может так случиться, что выбранный современный пакет просто не рассчитан на столь древнее оборудование.

Приемлемые результаты могут быть получены при использовании 24-игольчатого матричного принтера типа Epson LQ1050 (стоимостью порядка 600 долларов), хотя предпочтительнее использовать чернильный принтер типа Canon BJ-300/330, несмотря

на то, что последний стоит дороже. Однако надо помнить, что тот и другой работают довольно медленно, поскольку используют не текстовый, а графический режим. Для получения высококачественных распечаток или когда важна скорость печати, необходим лазерный принтер, хотя он стоит не менее тысячи долларов.

При использовании матричного принтера получающаяся распечатка не высока по качеству, и в уважающей себя типографии ее скорее всего не возьмут в качестве оригинала для печати, скажем, газеты. Навряд ли возможен поход в типографию с дискетой, на которой записан подготовленный в пакете DTP материал, поскольку чрезвычайно редко (тем более у нас) в типографии можно встретить устройства, позволяющие преобразовать информацию с дискеты, записанной на персональном компьютере, в форму, пригодную для последующей обработки на типографских печатных машинах. Можно, конечно, с этой дискетой пойти к приятелю, у которого есть такой же пакет DTP и лазерный принтер. Как правило, распечатка с лазерного принтера удовлетворяет работников типографии, если в этой распечатке нет особенно изысканных мест типа включенных в текст фотографий.

Однако прежде чем идти к приятелю, постарайтесь выяснить поточнее, чем он все-таки обладает. Дело в том, что сегодня среди персональных компьютеров существует два основных (и не совместимых между собой) семейства: компьютеры, изготовленные по стандартам фирмы IBM (их называют IBM-совместимыми), и компьютеры, изготовленные по стандартам фирмы Apple (их, чаще всего, называют МАКами). По этой причине существуют, естественно, пакеты DTP, разработанные для тех и других машин. С ними все понятно: либо одно, либо другое. Но наиболее мощные и популярные пакеты типа Ventura Publisher и PageMaker существуют в версиях как для одних, так и для других компьютеров. Поэтому возможен вариант, что вы работаете в пакете PageMaker на IBM-совместимом компьютере и получите утвердительный ответ от приятеля, что у него есть такой же пакет, а потом окажется, что он работает на Маке.

Но даже если вы не можете найти никого, у кого был бы такой же пакет DTP, еще не все потеряно. Не исключено, что ваш приятель сможет распечатать ваш материал, если вы передадите его на дискете в виде PostScript-файла. PostScript — это компьютерный язык, разработанный специально для описания страниц издания, готовящихся в пакетах DTP. Этот язык "понимаем" рядом современных принтеров, поэтому не исключено, что такой принтер сумеет распечатать ваш файл без помощи компьютера (если, конечно, вам удастся переправить его в принтер).

Но заметьте еще одну вещь: PostScript-файл, содержащий сложную страницу (не говоря уже о нескольких страницах), очень велик по размеру и может просто не поместиться на дискету.

Этот текст набран кеглем 8 пунктов

Этот текст набран кеглем 14 пунктов

Этот текст набран кеглем 8 пунктов

Этот текст набран кеглем 14 пунктов

Образцы текстов, отпечатанных на матричном и лазерном принтере.

Создание страницы

Получение распечатанной копии страницы, созданной в пакете DTP, — это конечный результат, но прежде ее надо создать. К сожалению, пакеты DTP слишком удобны и позволяют очень быстро создать плохой макет страницы. И даже при наличии самого современного оборудования и программного обеспечения можно получить результат, чтение которого доставит много горя читателю и вызовет насмешку у любого профессионала. Большинство профессиональных верстальщиков, работая над макетом, автоматически следуют нескольким простым, но важным правилам подготовки издания. Вот наиболее важные из них.

1. Не используйте на одной странице слишком много видов гарнитур (кстати, гарнитурой называется форма начертания всех букв алфавита, цифр и синтаксических символов): максимум три разных гарнитуры и не более чем трех размеров. В противном случае страница станет просто нечитаемой.
2. Если вы работаете над многостраничным или периодическим изданием, выберите один раз набор гарнитур и строго держитесь его в будущем. Так, например, журнал КомпьютерПресс постоянно набирается гарнитурами Таймс и Журнальная рубленная.
3. При работе над многостраничным, а тем более — над периодическим изданием, строго придерживайтесь сетки-схемы размещения текста и иллюстраций на странице, разработанной вашим художественным редактором.
4. Не стремитесь максимально заполнить текстом все пространство страницы: комната, битком заставленная мебелью, теряет свою привлекательность. Оставляйте побольше места под заголовки, поля и иллюстрации.

Очень важным аспектом при разработке макета страницы является правильный выбор набора гарнитур.

Гарнитуры разделяются на четыре широких категории. Первая из них включает гарнитуры типа Serif (Сериф — шрифт с засечками). Эти гарнитуры отличаются тем, что по краям основных штрихов литеры имеют засечки — небольшие “хвостики”. Вторая категория — это гарнитуры типа Sans serif (Санс сериф, полиграфисты называют ее иногда гротесковым шрифтом), представляющие собой “гладкие” литеры, лишенные каких-либо “хвостиков”, завитков и пр. Наиболее распространенными гарнитурами типа Serif являются Times (Таймс) и Литературная, а типа Sans serif — Helvetica (Гельветика). В некоторых пакетах DTP эти гарнитуры так и называются — Serif и Sans.

Третья категория — это гарнитуры типа Script (Скрипт — имитирует рукописный текст), которые используются в первую очередь для выделения какого-то текста. Примером такой гарнитуры является Zapf Chancery (Запф ченсири). Ею снабжено большинство лазерных принтеров, работающих с языком PostScript. Эта гарнитура хороша для печати, например, поздравительных посланий, но если этой гарнитурой набрать целую страницу, то читаться она будет с трудом.

Последняя категория включает гарнитуры типа “Pi“ (Пи), состоящие из специальных символов, а не привычных букв и цифр. Примером такой гарнитуры является Zapf Dingbats (Запф Дингбэтс), она включает квадратики, треугольники, стрелки, звездочки и пр. Такие гарнитуры разрабатываются для специальных применений, например для записи нот, формул химических соединений.

Некоторые гарнитуры включают только одно подмножество начертаний, другие же могут быть сгруппированы в некоторые семейства, такие, как italic (курсив), light (светлый), medium (нормальный), bold (полужирный), extra bold или enhanced (жирный), condensed (сжатый). Что представляет из себя каждое семейство, видно из его названия. Английские названия мы привели только для того, чтобы пользователь, осваивающий тот или другой пакет, понимал, о чем идет речь.

Каждая гарнитура может меняться по размерам. Определенный размер гарнитуры называется кеглем. Размер кегля измеряется в пунктах. Один типографский пункт — это 1/72 дюйма. В некоторых пакетах размер кегля указывается в пиках; одна пика равна 12 пунктам, а в одном дюйме, следовательно, насчитывается шесть пик. Гарнитура определенного размера, толщины и вида (курсив или прямой) называется шрифтом. Например, Таймс — это гарнитура, а Таймс прямой нормальный 9 пунктов — это конкретный шрифт, которым набрана изрядная доля журнала КомпьютерПресс.

Обычно пакеты настольных издательств, как и текстовые процессоры, поставляются с небольшим количеством гарнитур, явно недостаточным для серьезной работы. Вы можете докупать и устанавливать у себя (это несложно)

Times-Roman
Bookman
Helvetica
AvantGarde
Zapf Chancery
✇ ✈ ✓ ✔ ✗ ✖ ✘ ✚ ✎
✥ ✪ ☆ ✫ ✴ ✳ ✺ ✻ ✱ ✡

Примеры четырех основных категорий гарнитур. Сверху вниз: Сериф, Санс Сериф, Скрипт и Пи (шрифт со спецсимволами — в данном случае Zapf Dingbats).

как полные гарнитуры, так и отдельные шрифты. На Западе цена отдельной гарнитуры колеблется от 20 до 300 долларов (в зависимости от ее сложности). У нас одна гарнитура стоит несколько тысяч рублей. Наиболее известными разработчиками шрифтов на Западе являются фирмы Adobe, Monotype, International Typeface Corporation (ITC) и Agfa-Compugraphic, у нас в первую очередь назовем ПараГраф и Демос.

Что нужно купить

Для работы в пакете настольного издательства потребуется несколько больше оборудования и программного обеспечения, чем для игры в Тетрис. Точная конфигурация аппаратной части зависит от требований программного обеспечения, а его выбор зависит, во-вторых, от имеющегося у вас оборудования (если вы стеснены в средствах), и, во-первых, от ваших целей, от того, что именно вы собираетесь делать на этом пакете. Пакетов DTP разработано не меньше, чем сортов колбасы; самые дешевые стоят порядка 200 долларов, самые дорогие — свыше 2000. В нашей стране до Нового года русифицированный пакет Ventura Publisher фирмы Xerox продавался за 40 тыс. рублей, но это было до Нового года...

Этот пакет, наряду с пакетом PageMaker фирмы Aldus, является несомненным лидером в области подготовки обширных материалов, например книг и толстых журналов. Есть пакеты для разработки географических карт, подготовки статей по математике, химии и пр. В ближайших номерах мы дадим максимально полный обзор существующих пакетов DTP, пока же поговорим о другом.

Как правило, пакеты DTP не используются для набора текста и создания графических изображений — для этого существуют текстовые процессоры и специальные графические пакеты типа Publisher PaintBrush и Corel Draw. Выбирая тот или иной пакет DTP, проверьте, существует ли в нем возможность импорта (переноса) файлов из уже имеющихся у вас пакетов. В противном случае, потратив, скажем, 1000 долларов на покупку пакета DTP, вы вынуждены будете докупать текстовый процессор, совместимый с купленным пакетом DTP. Конечно, существуют специальные программы, позволяющие преобразовать форматы файлов, подготовленных в текстовых процессорах, в форматы пакетов DTP, но это, опять же, потребует дополнительных расходов и времени.

Пакеты настольных издательств существуют для всех стандартных сред персональных компьютеров: DOS, GEM, Microsoft Windows, Macintosh, UNIX. Почти все современные издательские программы работают с системой разворачивающихся меню, управляемой мышкой, видимо, это в какой-то степени является отражением успеха среды Windows, хотя наиболее популярные программы

Графические файлы

Издательские пакеты обладают возможностью обработки двух видов графики: векторной и точечной.

Точечные изображения, например, отсканированные фотографии, в простейшем случае являются не более чем сеткой, в узлах которой точки либо присутствуют (темный фон), либо отсутствуют (светлый тон). В этом случае изображение не может быть изменено по масштабу, поскольку при увеличении или уменьшении изображения произойдет попросту увеличение или уменьшение каждой точки, и качество снимка исчезнет.

В более сложном случае к каждой точке привязывается информация о том, каким оттенком серого цвета от практически белого до практически черного она обладает; в этом случае снимок может быть не только изменен по масштабу, но возможна его обработка, в частности, изменение контрастности и яркости.

Векторные изображения описываются математическим путем: нарисовать прямую от точки такой-то до точки такой-то, нарисовать окружность с центром в такой-то точке и радиусом X и т.п., поэтому они, естественно, могут быть изменены по масштабу до нужного размера.

Цвет в DTP

Цвет в DTP - это довольно новая штука, и сегодня лишь немногие пакеты (в частности, последние версии Ventura и PageMaker) поддерживают обработку цветных изображений.

Подбор нужного цвета в издательских программах осуществляется двумя путями. Первый - простой: те, кто когда-либо был связан с типографией, знают о существовании системы выбора цвета Pantone (на 1/6 суши аналогом является система "Радуга"), которая представляет из себя веер, состоящий из пластинок разного цвета, на каждой из которой стоит порядковый номер и указана интенсивность соответствующего цвета. Печатнику просто сообщают порядковые номера красок и соотношение, в котором он должен их смешать, чтобы получить нужный оттенок. Так же сделано и в издательских пакетах. Верстальщик выбирает по имеющейся шкале цветов нужные краски, вводит их коды, и принтер распечатает соответствующий участок страницы краской того или иного кода. Эта система позволяет на черно-белом фоне в границах выделенного участка отпечатать в одну краску текст, иллюстрацию или график.

Второй путь - это то, что в полиграфии называется "цветоделение". Цвет пятна в такой системе может быть определен как процентное соотношение четырех основных цветов: голубого, желтого, пурпурного и черного. В других издательских пакетах цвет пятна определяется как процентное соотношение красного, синего и зеленого.

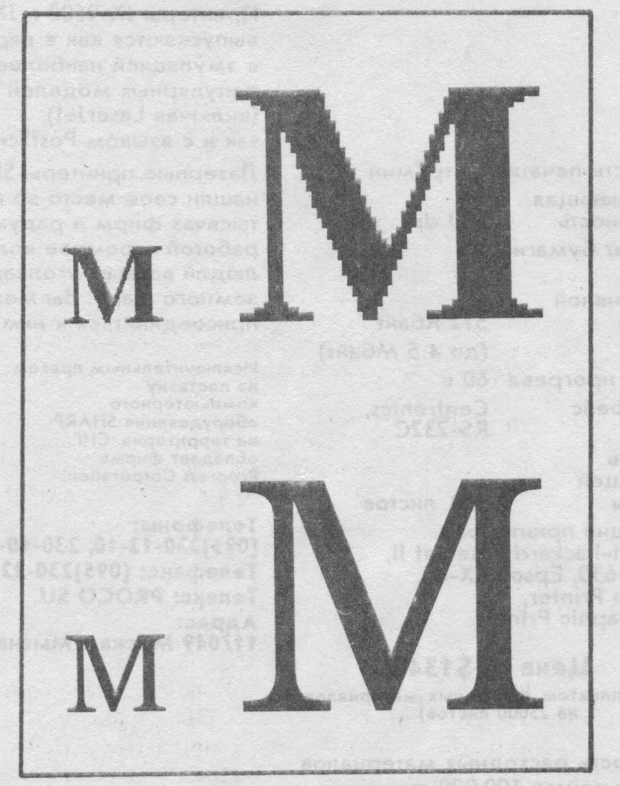
При подготовке к печати в такой системе для каждого из четырех цветов готовится своя пленка. Эти пленки, если очень постараться, могут быть изготовлены на хорошем черно-белом лазерном принтере. При объединении этих красок в типографии получается цветное изображение.

Шрифты

Все шрифты, как те, что появляются на экране монитора, так и те, что выводятся на принтер, могут быть разделены на две большие категории. Еще совсем недавно первая категория являлась и единственной – это так называемые матричные шрифты. Выбиралась матрица, скажем, восемь на восемь или шесть на двенадцать точек, и на ней, от точки к точке рисовался каждый символ создаваемого шрифта. Ближайший пример: таким способом каждый из нас рисует индекс на почтовых конвертах, только вместо линий, соединяющих соседние точки, в шрифте рисуется последовательность отдельных точек. Чем выше разрешение устройства, экрана или принтера, тем больше размер матрицы, а стало быть, и качество шрифта.

Еще не так давно стандартным разрешением экрана считалось 75 точек на дюйм, а лазерного принтера – 300 точек на дюйм. По этой причине матричный шрифт состоял, как правило, из двух частей: одна для экрана монитора, другая (матрица большего размера) для принтера. При отсутствии второй части издательская программа может попытаться растянуть экранный шрифт до размера матрицы принтера, но ни к чему хорошему, как видно из рисунка, это не приведет: буквы станут ступенчатыми и неуклюжими.

Ко второй категории шрифтов относятся масштабируемые шрифты. Они получили широкое распространение с созданием языка описания страниц PostScript. Начертание каждого символа описывается уравнениями как множество кривых. Когда в эти уравнения подставляется коэффициент размера шрифта, вводимый верстальщиком, тогда шрифт и "обретает" свой конкретный размер.



В отличие от масштабируемого шрифта (внизу), увеличение по размерам матричного шрифта приводит к искажению его внешнего вида.

типа Ventura Publisher фирмы Xerox или DeskPress фирмы GST по-прежнему выпускаются и для среды GEM.

Компьютеры

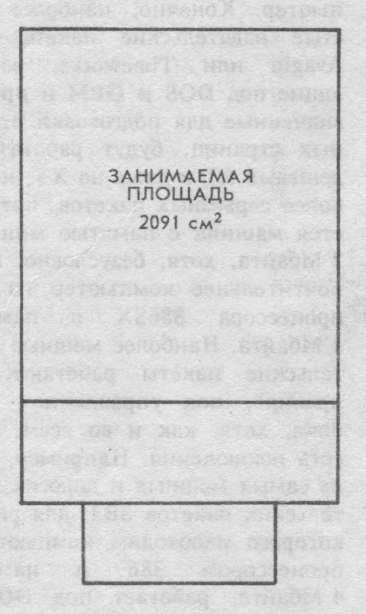
Отличительная черта пакетов DTP — активная работа с графикой, то есть с очень большими объемами информации, поэтому для нормальной работы вам потребуется довольно мощный компьютер. Конечно, наиболее простые издательские пакеты типа Avagio или Timeworks, работающие под DOS и GEM и предназначенные для подготовки отдельных страниц, будут работать на дешевых AT и даже на XT, но для более серьезных пакетов потребуется машина с памятью минимум 2 Мбайта, хотя, безусловно, предпочтительнее компьютер на базе процессора 386SX с памятью 4 Мбайта. Наиболее мощные издательские пакеты работают, как правило, под управлением Windows, хотя, как и во всем, здесь есть исключения. Например, один из самых мощных и дорогих издательских пакетов 3B2, для работы которого необходим компьютер с процессором 386 и памятью 4 Мбайта, работает под DOS, а весьма простые издательские пакеты для начинающих — Publisher фирмы Microsoft и PagePlus фирмы Serif — работают только в среде Windows.

Вы должны помнить, что одна отсканированная страница формата A4 потребует порядка 1 Мбайта дисковой памяти (если это не полутоновая картинка), и, стало быть, если ваша работа связана со сканированием большого числа изображений, вам потребуется жесткий диск емкостью не менее 80 Мбайт. Следует иметь в виду, что при записи 256 оттенков серого размер файла достигнет 8 Мбайт, а при записи еще и цвета — она уже утроится. Кроме того, чтобы спокойно работать с такими огромными массивами данных, графические пакеты часто хранят промежуточные изображения во временных файлах на том же винчестере. Это может еще увеличить необходимый объем винчестера.

Другая покупка, сравнимая по цене с подержанным автомоби-

НАЙДИТЕ ОТЛИЧИЯ

HP LaserJet-IIP



Скорость печати	4 стр/мин
Разрешающая способность	300 dpi
Формат бумаги	A4
Объем оперативной памяти	512 Кбайт (до 4.5 Мбайт)
Время прогрева	55 с
Интерфейс	Centronics, RS-232C
Емкость подающего лотка	50 листов

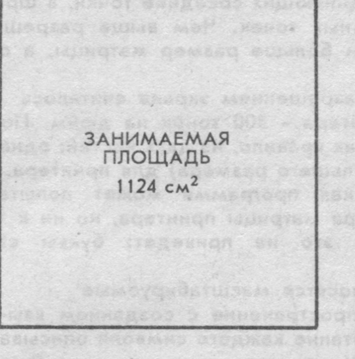
Цена $1450*
(с комплектом расходных материалов на 4000 листов)

Дополнительная кассета для подачи бумаги емкостью 250 листов
Цена $199*

Общая цена $1649*

Стоимость расходных материалов для печати 100,000 страниц
$2200*

SHARP JX-9500E



Скорость печати	6 стр/мин
Разрешающая способность	300 dpi
Формат бумаги	A4
Объем оперативной памяти	512 Кбайт (до 4.5 Мбайт)
Время прогрева	60 с
Интерфейс	Centronics, RS-232C
Емкость подающей кассеты	250 листов

Эмуляция принтеров Hewlett-Packard LaserJet II, Diablo-630, Epson FX-80, IBM Pro Printer, IBM Graphic Printer

Цена $1340
(с комплектом расходных материалов на 25000 листов)

Стоимость расходных материалов для печати 100,000 страниц
$677

Лазерный принтер стал обязательным атрибутом современного офиса. Качество печати и скорость работы делают его незаменимым, если вы стремитесь к грамотному подходу к бизнесу.

Лазерные принтеры SHARP — это предельная экономия средств при эксплуатации. Еще одно важное качество — компактность. Принтер SHARP сэкономит место в Вашем офисе.

Высокое качество работы принтеров SHARP особенно важно для настольных издательских систем.

Принтеры JX-9500 и JX-9700 выпускаются как в варианте с эмуляцией наиболее популярных моделей (включая LaserJet), так и с языком PostScript.

Лазерные принтеры SHARP нашли свое место во многих тысячах фирм и радуют своей работой огромное количество людей во всех уголках земного шара. Вы можете присоединиться к ним.

Исключительным правом на поставку компьютерного оборудования SHARP на территории СНГ обладает фирма Progress Corporation.

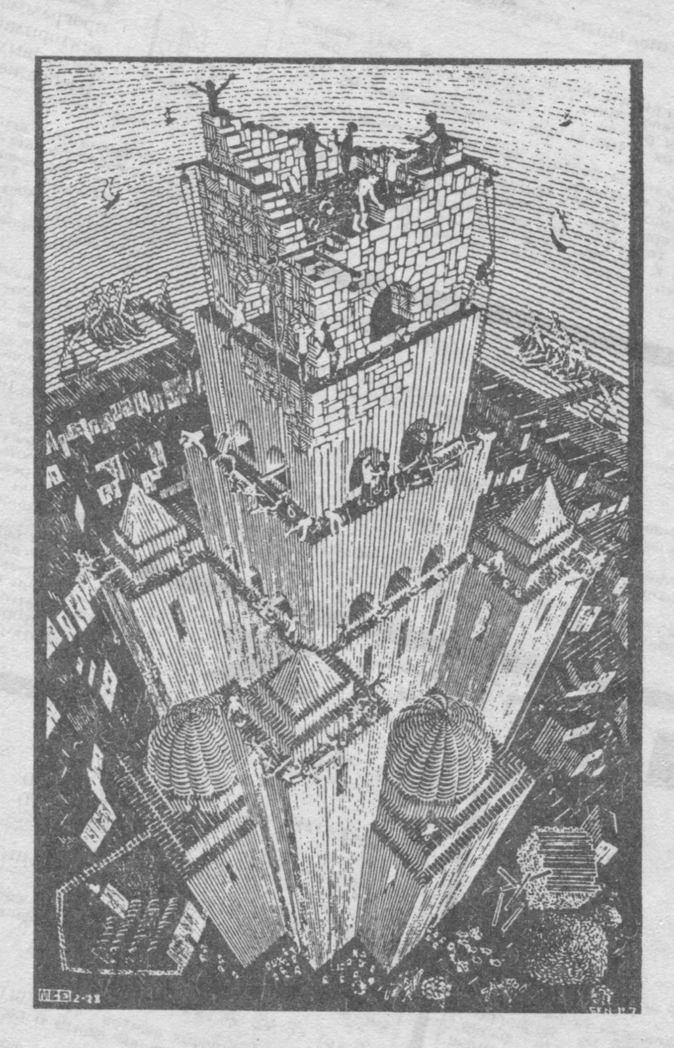
Телефоны: (095)230-12-10, 230-10-75
Телефакс: (095)230-22-26
Телекс: PROCO SU
Адрес: 117049 Москва, Мытная 1-30

SHARP

* Цены приведены по состоянию на 31.01.92 у фирмы "Стиплер" — дилера Hewlett-Packard

лем, — это монитор высокого разрешения формата А3 или в крайнем случае А4 (с вертикальным расположением экрана). Это дороже среднего компьютера, зато вам не придется ломать глаза, пытаясь разобрать на экране, что же такое вы создали в своем издательском пакете.

В общем, если вся ваша издательская деятельность будет заключаться в подготовке писем и двухстраничных отчетов, сэкономьте свои деньги и ограничьтесь скромной АТишкой. Но ежели вы собираетесь издавать Большую Энциклопедию СНГ в 52 томах, то при покупке компьютера заплатите пару лишних юаней. Ведь скупой платит дважды...



Сканеры

Сканер позволяет создавать электронную копию изображения для последующей ее обработки в издательском пакете. Чтобы получить качество, сравнимое с газетным, потребуется настольный формата А4 или ручной сканер, обеспечивающий разрешение 300 точек на дюйм (лучше еще больше) и передающий 256 оттенков серого цвета. Такие сканеры позволяют получить черно-белые копии с цветных и черно-белых фотографий и рисунков, которые после обработки можно будет распечатать на принтере.

Существуют как ручные, так и настольные цветные сканеры, позволяющие считывать цветные изображения, которые могут быть затем распечатаны на цветном принтере. Однако здесь существуют два нюанса. Во-первых, это весьма и весьма дорогое оборудование, во-вторых, довольно мало издательских пакетов по-настоящему поддерживает на персональных работу с цветом. Из известных нам пакетов такими являются Ventura 4.0 и QuarkXPress для Windows. Лучшими фирмами-производителями сканеров считаются Hewlett-Packard и Microtec. Хорошие цветные сканеры делают фирмы SHARP, Varityper, Epson, теперь еще и Hewlett-Packard.

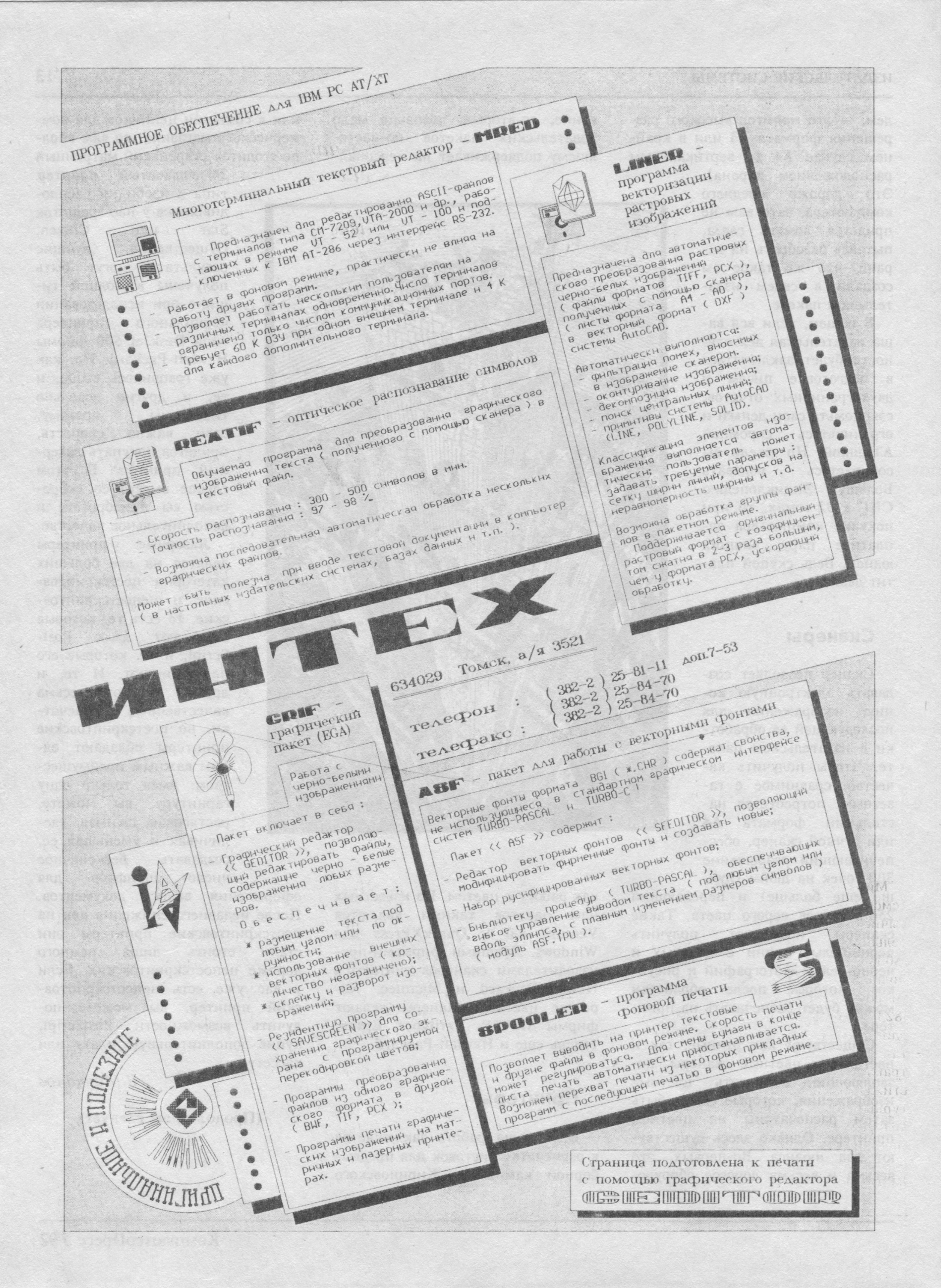
Принтеры

Если ваша работа заключается в распечатке листовок для предвыборной кампании Жириновского или в создании ценников для коммерческого магазина, то вам вполне сгодится старенький матричный 24-игольчатый принтер типа в изобилии расплодившихся у нас трещоток Star или Citizen. Существенно лучшие результаты могут быть получены в полной тишине при использовании чернильного принтера типа DeskJet 500 фирмы Hewlett-Packard. Но, как уже говорилось выше, и те, и другие довольно тихоходны, поэтому, если важна скорость, придется покупать лазерный принтер. В этом случае вместе со скоростью вы приобретете и дополнительное качество.

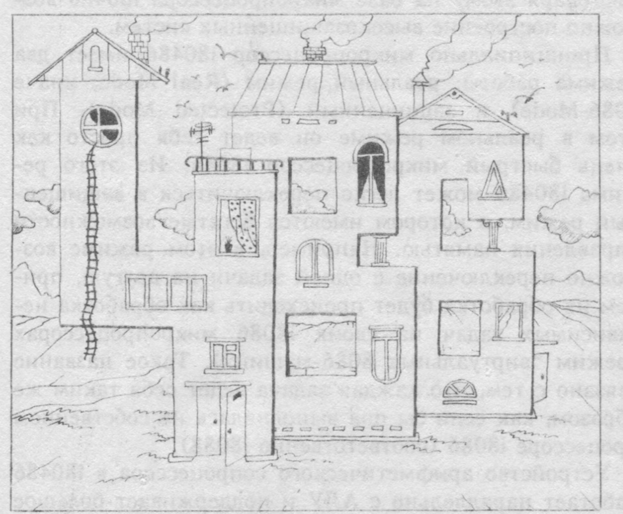
Лазерные принтеры делятся на две больших категории: постскриптовские и непостскриптовские, то есть те, которые понимают язык PostScript, и те, которые его не понимают. И те, и другие выдают весьма качественные распечатки, но постскриптовские принтеры обладают одним важным преимуществом: имея только одну гарнитуру, вы можете, растягивая, сжимая, увеличивая и уменьшая ее, создавать бесконечное число шрифтов для оформления ваших документов. После недавнего снижения цен на постскриптовские принтеры они стали стоить лишь немного дороже непостскриптовских. Если у вас уже есть непостскриптовский принтер, вы можете получить возможности PostScript, купив дополнительную плату или кассету.

Б.Молчанов

(Продолжение следует)



ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ IBM PC AT/XT
Многотерминальный текстовый редактор - MRED
Предназначен для редактирования ASCII-файлов с терминалов типа СМ-7209, VTA-2000 и др., работающих в режиме VT - 52 или VT - 100 и подключенных к IBM AT-286 через интерфейс RS-232.
Работает в фоновом режиме, практически не влияя на работу других программ.
Позволяет работать нескольким пользователям на различных терминалах одновременно. Число терминалов ограничено только числом коммуникационных портов.
Требует 60 К ОЗУ при одном внешнем терминале и 4 К для каждого дополнительного терминала.
LINER - программа векторизации растровых изображений
Предназначена для автоматического преобразования растровых черно-белых изображений (файлы форматов TIFF, PCX), полученных с помощью сканера (листы формата A4 - A0), в векторный формат (DXF) системы AutoCAD.
Автоматически выполняются:
- фильтрация помех, вносимых в изображение сканером;
- оконтуривание изображения;
- декомпозиция изображения;
- поиск центральных линий;
- примитивы системы AutoCAD (LINE, POLYLINE, SOLID).
Классификация элементов изображения выполняется автоматически; пользователь может задавать требуемые параметры: сетку ширин линий, допусков на неравномерность ширины и т.д.
Возможна обработка группы файлов в пакетном режиме.
Поддерживается оригинальный растровый формат с коэффициентом сжатия в 2-3 раза большим, чем у формата PCX, ускоряющий обработку.
REATIF - оптическое распознавание символов
Обучаемая программа для преобразования графического изображения текста (полученного с помощью сканера) в текстовый файл.
- Скорость распознавания : 300 - 500 символов в мин.
- Точность распознавания : 97 - 98 %.
- Возможна последовательная автоматическая обработка нескольких графических файлов.
Может быть полезна при вводе текстовой документации в компьютер (в настольных издательских системах, базах данных и т.п.).
ИНТЕХ
634029 Томск, а/я 3521
телефон : (382-2) 25-81-11 доп.7-53
телефакс : (382-2) 25-84-70
(382-2) 25-84-70
GRIF - графический пакет (EGA)
Работа с черно-белыми изображениями
Пакет включает в себя :
Графический редактор << GEDITOR >>, позволяющий редактировать файлы, содержащие черно - белые изображения любых размеров.
Обеспечивает :
* размещение текста под любым углом или по окружности;
* использование внешних векторных фонтов (количество неограничено);
* склейку и разворот изображений;
- Резидентную программу << SAVESCREEN >> для сохранения графического экрана с программируемой перекодировкой цветов;
- Программы преобразования файлов из одного графического формата в другой (BWF, TIF, PCX);
- Программы печати графических изображений на матричных и лазерных принтерах.
ASF - пакет для работы с векторными фонтами
Векторные фонты формата BGI (*.CHR) содержат свойства, не использующиеся в стандартном графическом интерфейсе систем TURBO-PASCAL и TURBO-C !
Пакет << ASF >> содержит :
- Редактор векторных фонтов << SFEDITOR >>, позволяющий модифицировать фирменные фонты и создавать новые;
- Набор русифицированных векторных фонтов;
- Библиотеку процедур (TURBO-PASCAL), обеспечивающих гибкое управление выводом текста (под любым углом или вдоль эллипса, с плавным изменением размеров символов) (модуль ASF.TPU).
SPOOLER - программа фоновой печати
Позволяет выводить на принтер текстовые и другие файлы в фоновом режиме. Скорость печати может регулироваться. Для смены бумаги в конце листа печать автоматически приостанавливается. Возможен перехват печати из некоторых прикладных программ с последующей печатью в фоновом режиме.
ПРИЛИЧНЫЕ И ПОЛЕЗНЫЕ
Страница подготовлена к печати с помощью графического редактора GEDITOR



Интегральная схема микропроцессора i80486 была анонсирована фирмой Intel в 1989 году на выставке Comdex в Чикаго. И хотя, очевидно, нам не долго уже осталось ждать появления нового 586-го микропроцессора, на сегодняшний день одним из самых мощных универсальных микропроцессоров является i80486.

Архитектура микропроцессора i80486

"В архитектуре взаимосвязаны функциональные, технические и эстетические начала — польза, прочность, красота."

(Из словаря)

Микропроцессор i80486 достойный продолжатель семейства i80x86, поскольку он позволяет создавать вычислительные системы намного более производительные, чем, предположим, на микропроцессоре i80386. В принципе микросхема i80486 состоит из полного микропроцессора i80386, арифметического сопроцессора i80387 и контроллера кэш-памяти i82385. Понятно, что речь не идет о механическом соединении функциональных устройств на одном кристалле. Высокая степень интеграции (более миллиона транзисторов) — это не просто технологическое достижение, а возможность реализовать новые архитектурные решения, которые позволяют повысить производительность микропроцессора. Так, например, в микропроцессоре i80486 используется конвейерная обработка команд, достаточно широко применяемая в RISC-процессорах. При одной и той же тактовой частоте с i80386 микропроцессор i80486 работает в три-четыре раза производительнее. Это связано с уменьшением количества циклов (тактов), необходимых для выполнения команды (инструкции).

Три в одном

На рис. 1 приведена примерная блок-схема микропроцессора i80486, на которой достаточно легко выделить его основные функциональные узлы — устройство управления памятью (Memory Managment Unit, MMU), модуль - арифметического сопроцессора (Floating Point Unit, FPU) и блок кэш-памяти (Cache Unit). Несмотря на то, что микропроцессор i80486 обладает всеми свойствами, присущими i80386, имеются, конечно, и различия. Например, в набор команд микропроцессора i80486 наряду с известными инструкциями для i80386, включено несколько дополнительных команд. Однако устройство управления памятью MMU полностью совместимо с аналогичным устройством в i80386. То же можно сказать и о FPU. А это, вообще говоря, означает, что все программное

обеспечение, работающее на серии микропроцессоров 80x86/80x87, без внесения каких-либо изменений будет работать и на i80486.

Устройство управления памятью MMU состоит в свою очередь из устройства сегментации (Segmentation Unit) и устройства управления страницами (Paging Unit). Благодаря делению памяти на сегменты (сегментации) коды программ и области данных могут находиться практически в любом месте логического адресного пространства. Иными словами, при сегментной организации памяти каждый модуль занимает свою собственную сплошную область памяти, тогда как при страничной организации модуль разбивается на отдельные страницы. В свою очередь, устройство управления страницами позволяет реализовать механизм страничной организации памяти, который функционирует на более низком уровне и является прозрачным по отношению к процессу сегментации. Это дает возможность управления физическим адресным пространством. Следует также отметить, что страничная организация памяти (paging) может быть отключена чисто программным способом.

Каждый сегмент памяти может подразделяться на один или несколько 4-Кбайтных блоков. А память, в свою очередь, может быть организована в виде одного или нескольких таких сегментов, размер которых может достигать 4 Гбайт. Причем каждому такому сегменту могут придаваться соответствующие атрибуты, которые определяют его расположение в памяти, величину, вид (стек, код, данные) и характеристику защиты. Каждая задача, выполняемая на i80486, может в принципе состоять из 16 381 сегмента, по 4 Гбайта каждый. Таким образом, диапазон виртуальной адресации достигает 64 Тбайт. Устройство сегментации микропроцессора поддерживает четыре механизма защиты приложений и системного режима друг от друга. Благодаря этому на базе микропроцессора i80486 возможно построение высокозащищенных систем.

Принципиально микропроцессор i80486 имеет два режима работы: реальный режим (Real Mode, иначе 8086-Mode) и защищенный (Protected Mode). При этом в реальном режиме он ведет себя просто как очень быстрый микропроцессор i8086. Из этого режима i80486 может легко переключиться в защищенный режим, в котором имеются богатые возможности управления памятью. Например, в этом режиме возможно переключение с одной задачи на другую, причем их обработка будет происходить как обработка независимых задач на своих i8086 микропроцессорах (режим "виртуальных 8086-машин"). Такое название связано с тем, что каждая задача ведет себя таким же образом, как если бы она выполнялась на собственном процессоре i8086 (соответственно i8088).

Устройство арифметического сопроцессора в i80486 работает параллельно с АЛУ и поддерживает большое количество арифметических команд для различных типов операндов. Это, в частности, многочисленные трансцендентные функции, такие как синус, косинус, тангенс, логарифм и т.д. Нужно отметить, что FPU полностью поддерживает стандарт ANSI/IEEE 754-1985 на арифметические операции с плавающей точкой.

Размер внутренней кэш-памяти микропроцессора i80486 составляет 8 Кбайт. Сразу следует сказать, что для некоторых приложений такой размер кэш-памяти вовсе не достаточен. Это касается в первую очередь САПР и настольных издательских систем. Однако в этом случае при разработке соответствующей системы можно воспользоваться кэш-контроллером i82485 и внешней кэш-памятью объемом 64 или 128 Кбайт. Внутренняя кэш-память выполнена как ассоциативная и работает в режиме write through (то есть данные заносятся одновременно и в кэш-память, и в ОЗУ). Кэш



Рис. 1

в микропроцессоре i80486 безусловно имеет и некоторые особенности. В частности, страницы памяти могут быть обозначены либо как "кэшируемые", либо как "некэшируемые", причем такое обозначение можно вводить как аппаратно, так и программно. Впрочем, даже само устройство кэша может быть также отключено как аппаратно, так и программно. По некоторым оценкам, использование встроенной кэш-памяти экономит до 80% времени, необходимого для обращения к ОЗУ.

Регистры, регистры, регистры...

Микропроцессор i80486 содержит, конечно, все регистры, имеющиеся в микросхемах i80386 и i80387. Регистры рассматриваемого микропроцессора можно условно подразделить на три большие группы: базовые регистры, регистры системного уровня и регистры

Регистры общего назначения

31	15	7	0	
	AH	AX	AL	EAX
	BH	BX	BL	EBX
	CH	CX	CL	ECX
	DH	DX	DL	EDX
		SI		ESI
		DI		EDI
		BP		EBP
		SP		ESP

Сегментные регистры

15	0
CS	
SS	
DS	
ES	
FS	
GS	

Указатель команд EIP

31	15	0
	IP	

Регистр флагов EFLAGS

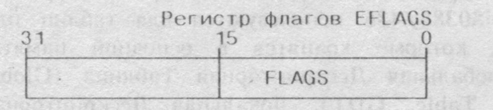


Рис.2

Биты регистра EFLAGS

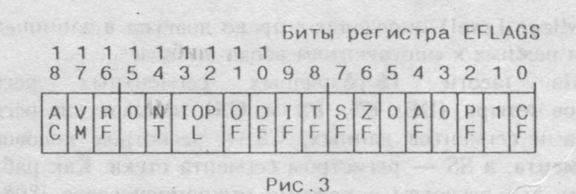


Рис.3

плавающей точки. Рассмотрим эти группы регистров несколько подробнее.

Программная модель i80486 определяется главным образом базовыми регистрами, которые приведены на рис. 2. Это — восемь 32-разрядных регистров общего назначения, которые могут работать с операндами данных размерами 1, 2 и 4 байта или полями бит от 1 до 32. Однако операнды адреса могут иметь длину только либо 2, либо 4 байта. Эти 32-разрядные регистры обозначаются как EAX, EBX, ECX, EDX, ESI, EDI, EBP и ESP. Обозначения двух младших байт этих регистров пользователям микропроцессоров i8088/86/286 уже хорошо известны. Конечно, это — AX, BX, CX, DX, SI, DI, BP и SP. Причем следует отметить, что при доступе к старшим двум байтам этих 32-разрядных регистров содержимое двух младших байтов не изменяется. Этим обуславливается совместимость сверху-вниз с микропроцессором i80286. Операции с байтовыми операндами (старший и младший байты) доступны только для четырех регистров — AX, BX, CX и DX, причем старшие байты соответствуют обозначениям AH, BH, CH и DH, а младшие — AL, BL, CL и DL.

Регистр указателя команд (Instruction Pointer, EIP) — также 32-разрядный — указывает на следующую выполняемую команду, конечно же, косвенно. Этот регистр указывает только смещение относительно начала кодового сегмента адреса, который образуется с использованием содержимого регистра CS и элемента дескрипторной таблицы (подробнее об этом несколько позже). Два младших байта регистра EIP соответствуют 16-разрядному регистру IP, который используется только при 16-битной адресации.

Для обозначения базового 32-разрядного регистра побитной индикации состояния процессора используется так называемый регистр флагов — EFLAGS. Этот регистр содержит биты, характеризующие последнюю выполненную процессором операцию, и некоторую другую служебную информацию. На рис. 3 приведены обозначения битов в регистре EFLAGS. Поясним назначение некоторых из них. Если в регистре EFLAGS установлен 17-й бит — VM (Virtual 8086 Mode), то микропроцессор i80486 переходит в режим "виртуальных 8086-машин". Конечно, это возможно только в защищенном режиме. Бит 16 — RF (Resume Flag) используется обычно для отладочных режимов. При его установке прерывание, вызванное фатальной ошибкой, игнорируется и выполняется переход на следующую инструкцию. Бит 14 — NT (Nested Task) используется в i80486 для наблюдения за несколькими прерванными или вызванными по инструкции CALL программами. Два бита 12 и 13 — IOPL (Input/Output

Privilege Level) определяют право доступа в защищенном режиме к инструкциям ввода-вывода.

Из шести 16-разрядных сегментных регистров четыре (DS, ES, FS и GS) называются регистрами сегментов данных, CS — регистром кодового сегмента, а SS — регистром сегмента стека. Как работают эти регистры в режиме микропроцессора i8086, достаточно хорошо известно. Для получения 20-разрядного физического адреса необходимо суммировать 16-битное смещение с базовым адресом, хранящимся в одном из сегментных регистров. Причем сложение происходит так, что содержимое сегментного регистра сдвинуто относительно второго слагаемого (смещения) влево на 4 разряда. То есть по сути происходит сложение 20-битового базового адреса (4 младших разряда которого, безусловно, всегда нули) с 16-битовым смещением, в результате, конечно, получается необходимый 20-разрядный адрес. При работе же микропроцессора i80486 (ну и, разумеется, i80386) в защищенном режиме исполнительный физический адрес образуется несколько иначе. Можно сказать, что он образуется из пяти компонент. Прежде всего отметим, что в защищенном режиме сегментные регистры называются селекторными, или просто селекторами. Селекторный регистр указывает на соответствующий элемент в одной из Дескрипторных Таблиц, расположенных в памяти. Причем базовый адрес этой Таблицы хранится либо в регистре Глобальной Дескрипторной Таблицы, либо в регистре Локальной Дескрипторной Таблицы, хотя в общем случае может храниться и в ином регистре микропроцессора. Каждый элемент Дескрипторной Таблицы содержит базовый адрес определенного сегмента, его величину и права доступа. У i80486 элемент Глобальной Дескрипторной Таблицы имеет размер 8 байт и состоит из 32-разрядного базового адреса, 20-разрядного поля размера сегмента и атрибутов дескриптора. В поле размера сегмента, например, может храниться либо

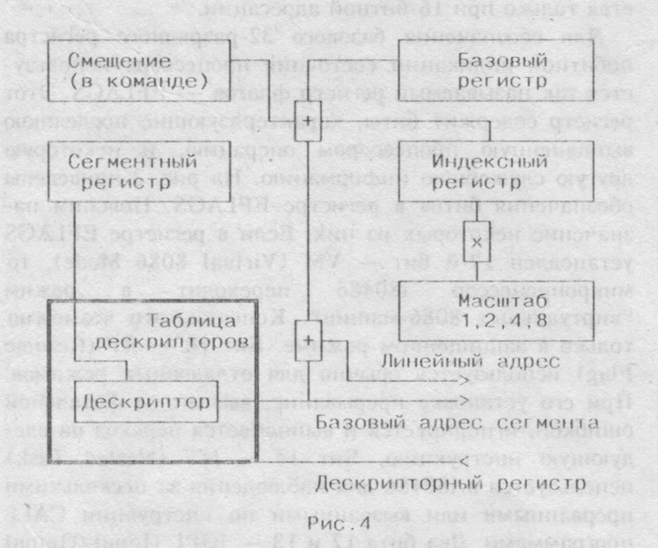


Рис. 4

количество байт (до 1 Мбайта), которое содержит данный сегмент, либо количество страниц по 4096 байт каждая. Следовательно, размер сегмента в защищенном режиме может варьироваться от 1 байта до 4 Гбайт. Таким образом, результирующий адрес в защищенном режиме может быть образован в общем случае суммированием содержимого базового регистра с содержимым индексного регистра, умноженного на масштабный коэффициент (1, 2, 4 или 8), плюс смещение в самой команде и плюс базовый адрес сегмента из элемента Дескрипторной Таблицы (рис. 4).

Микропроцессор i80486 имеет три управляющих регистра (Control Registers) системного уровня — CR0, CR2 и CR3 (регистр CR1 зарезервирован фирмой Intel для дальнейшего расширения). В регистре CR0 из имеющихся 32 бит для целей управления и определения статуса используется только 10 (рис. 5). Причем 5 из этих 10 бит появились только в i80486 (в i80386 их нет). Это биты — CD, NW, AM, WP И NE. Функционально все эти 10 бит характеризуют следующее: режимы микропроцессора i80486 — PG, PE, режимы встроенного устройства кэш-памяти — CD, NW, встроенное устройство управления плавающей арифметикой — TS, EM, MP и NE, управление контролем выравнивания — AM, защита записи супервизора — WP. Младшее слово регистра CR0 используется также в качестве статусного слова (слова состояния) микропроцессора (Machine Status Word, MSW), для того чтобы сохранить совместимость с защищенным режимом i80286. Следовательно, команды LMSW и SMSW (Load MSW, Store MSW) действуют только для нижних 16 бит CR0 и игнорируют новые биты. Для того, чтобы можно было работать с этими новыми битами, в набор команд микропроцессора i80486 включена специальная инструкция MOV CR0,Reg.

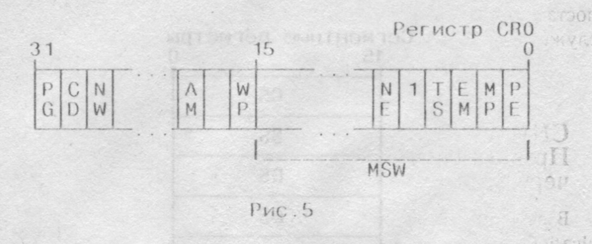


Рис. 5

В управляющем регистре CR2 находится адрес, по которому появилась ошибка при разбиении памяти на страницы. Управляющий регистр CR3 используется только тогда, когда в регистре CR0 установлен бит PG. В этом случае регистр содержит базовый адрес таблицы страниц, которые может использовать микропроцессор для каждой задачи. Сама же эта таблица всегда выровнена на одну страницу (4 Кбайта).

Четыре адресных системных регистра в микропроцессорах i80386/486 используются для таблиц или сегментов, которые хранятся в основной памяти. Это — Глобальная Дескрипторная Таблица (Global Descriptor Table, GDT), Локальная Дескрипторная Таблица (Local Descriptor Table, LDT), Дескрипторная Таблица Прерываний (Interrupt Descriptor Table, IDT)

и Сегмент Состояния Задачи (Task State Segment, TSS). Для обозначений соответствующих регистров используются аббревиатуры — GDTR, LDTR, IDTR и TR. Регистры GDTR и IDTR содержат 32-разрядный базовый адрес, а также 16-разрядную границу для таблиц GDT и IDT. Таблицы GDT и IDT находятся в распоряжении всех задач как глобальные сегменты, в то время как LDT и TSS зависят от конкретной задачи. В регистрах LDTR и TR хранятся 16-разрядные селекторы для выбора LDT- и соответственно TSS-дескриптора.

Набор регистров для операций с плавающей точкой включает в себя восемь 80-разрядных регистров данных, 48-разрядные указатели данных и команд, а также 16-разрядные регистры управления, состояния и слова признаков. Поскольку принцип работы FPU полностью соответствует работе i80387, а арифметические сопроцессоры — хорошая тема для отдельного разговора, подробно останавливаться на этих регистрах мы пока не будем.

В следующих выпусках нашего журнала мы обязательно вернемся к разговору о микропроцессорах фирмы Intel, и не только о них...

А. Борзенко

Сотовая связь в Москве

На этой неделе пришли три сообщения о том, как быстро становится реальностью повсеместное распространение сотовой связи. Кирилл Чащин пишет, что Московская Сотовая Связь объявила о начале опытной эксплуатации сотовой системы на частоте 450 Мгц — управляющий директор Вероника Бломштед ожидает, что через 5 лет у системы будет 60 000 клиентов. Партнером Московской Городской Телефонной Сети по МСС является компания US West (Колорадо, США). Даже если МСС не увенчается успехом, то есть еще инженеры с Вымпела, которые на этой неделе учатся в США у фирмы Plexsys, как создавать узлы сотовой связи для работы в России. Другая сделка Вымпела с компанией Cellular Inc. (Колорадо) может поставить в повестку дня создание службы PCN на микроволновой основе.

The Teleputing Hotline, January 13, 1991.

СНГ: Прямая связь через Sprint

В рамках соглашения с IDB Communications, которая получила право на эксплуатацию спутниковых линий вместе с AT&T, Sprint предлагает прямую телефонную голосовую связь из США с Россией и другими республиками бывшего Советского Союза. Звонки в эти страны из США через Intersputnik попадают на цифровую АТС в Ст.-Петербурге, откуда расходятся по стране.

Газета Die Welt подтвердила сообщение о том, что немецкие телефонные компании, включая государственную Deutsche Bundespost Telekom, ведут обсуждение с властями СНГ вопросов, касающихся предложений об участии в спутниковом проекте Romantis. Стоимость проекта может составить от 3 до 7 миллиардов марок (от 2 до 4,7 млрд. долл.). Romantis предполагает вывод 3 спутников на геостационарную орбиту, что даст возможность обрабатывать до 3-х миллионов звонков одновременно, при этом существенная часть полосы несущих частот будет предназначена для передачи данных. Немцы хотят в качестве страховки получить экспортные кредитные гарантии, составляющие до 2/3 стоимости проекта. Только Bundespost в этом году вложит в проект 160 миллионов марок.

Сеть на основе спутниковой связи будет легче и дешевле установить, чем наземные линии. Особенно важно то, что такую сеть можно установить быстрее. Приемные и передающие устройства с "тарелками" диаметром 1,5 метра в провинции и более крупные приемо-передающие "тарелки" для телефонных сетей крупных городов обеспечат быстрый доступ к национальным и международным линиям. После этого спутниковые сигналы могут быть перенаправлены в Германию по существующим сетям Bundespost.

The Teleputing Hotline, January 13, 1991.

Австралия: Война против модемов

Операторы BBS попали под "огонь", когда Austel, регулирующий орган Австралии в области телефонии, начала кампанию против "нелегальных" модемов, сообщает Newsbytes Пол Цукер. Кампания, продолжающаяся уже 2 месяца, направлена на модели HST американской фирмы Robotics, которые не отвечают требованиям Austel на изолированность линий, уровень сигнала и скорость набора. Austel утверждает, что они могут мешать разговорам на соседних линиях.

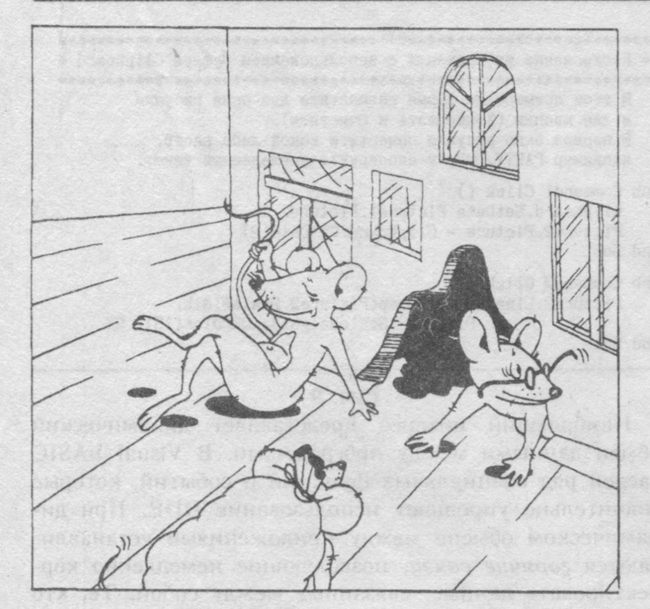
Austel определяет модемы с помощью метода Subscriber Automatic Line Tester (SALT). BBS стали объектом борьбы, когда Austel просмотрела список национальных узлов сети FidoNet (FidoNet является международной сетью электронных бюллетеней BBS) и заметила те из BBS, которые используют HST. Austel потребовала отсоединения модемов HST, предупредив, что следующей мерой будет штраф в 12 000 долларов. Одна американская компания в Сиднее получила предупреждение и убрала свой модем в одном офисе, но оставила в другом. В результате — штраф 80 000 долларов.

The Teleputing Hotline, January 13, 1991.

База данных по мошенничеству

Американские страховые фирмы объявили об инициативе по отслеживанию мошенничества в области страхования, когда люди специально разбивают свои машины, чтобы избежать их отдачи при невыплаченном кредите или сообщают, что они украдены, а на самом деле продают их за границу. Национальное Бюро по Преступности в Области Страхования (NICB) объединит Национальное Бюро по Кражам Автомобилей и Институт по Предупреждению Преступности в Области Страхования, при этом будет создана крупная компьютерная база данных, связанная с компьютерной системой ФБР. Спонсорами выступают более 700 компаний-страхователей, утверждающих, что местное законодательное решение вопроса их не устраивает. Они утверждают, что теряют около 20 миллиардов долларов в год. Группы борцов за гражданские права и свободы озабочены образованием связей между детективами частных фирм и досье ФБР, пишет корреспондент Newsbytes Джон Маккормик.

The Teleputing Hotline, January 13, 1991.



Фирма Microsoft первой сделала шаг навстречу многочисленному сообществу начинающих и непрофессиональных программистов, выпустив новый инструментальный пакет Visual BASIC. Этот продукт не только помогает вступающим на путь программирования понять идеологию среды Windows, но и дает возможность создавать свои собственные программы, не уступающие по характеристикам известным программам крупных фирм.

Microsoft Visual BASIC: первые шаги в программировании для Windows

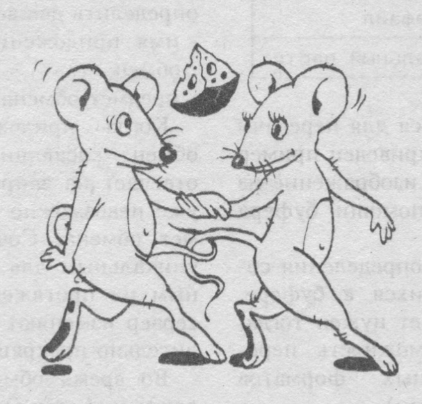
Обмен данными

В Windows имеется три стандартных возможности обмена данными между приложениями. Это статический обмен данными через буфер хранения, динамический обмен данными (DDE) и, наконец, связь и внедрение объектов (OLE). В Visual BASIC стандартно реализованы два из этих трех видов обмена — обмен через буфер хранения и динамический обмен данными.

Помимо этого, VB позволяет передавать и более простую информацию — последовательность нажатий на клавиши. Допустим, нам необходимо непосредственно из нашей программы активизировать некоторую другую программу, уже запущенную, но изображенную в виде иконки. Если бы у нас не было мыши, то мы поступили бы следующим образом: перевели бы фокус на иконку нужной программы (через меню Диспетчера Программ), затем нажали комбинацию клавиш Alt-Пробел (в результате чего появилось бы системное меню программы) и, наконец, нажали клавишу "R" (что привело бы к восстановлению изображения окна программы). Именно такую последовательность действий и надо воспроизвести в прикладной программе.

Окончание. Начало в №2'92.

Для перевода фокуса на другую программу используется функция AppActivate, активизирующая (то есть передающая фокус) другую, уже запущенную программу. Для имитации нажатий на клавиши в Visual BASIC применяется встроенная функция SendKeys. В качестве параметра этой функции указывается последовательность клавиш, помещенная в кавычки. Ряд клавиш, не имеющих графического представления (TAB, ENTER, ALT, SHIFT, CONTROL, F1...F12 и т.п.) имеют специальные обозначения. Так, описанная выше последовательность, восстанавливающая изображение окна, запишется в виде



```
SendKeys "%{ }R"
```

Если перед функцией SendKeys не была выполнена команда AppActivate, то вся последовательность будет направлена той программе, которая ее посылает.

Как я уже отметил, подобный обмен данными относится к самым элементарным и может быть использован только в ограниченных слу-

чаях. Наиболее распространенным способом обмена является обмен через *буфер хранения*. **Буфер хранения Windows** представляет собой достаточно универсальное средство обмена, позволяющее хранить текст, растровые изображения, Windows-мета-файлы и универсальные растры.

Для передачи данных через буфер хранения используется специальный объект — **Clipboard**. Этот объект не имеет свойств или событий, но к нему можно применить ряд методов, позволяющих осуществлять обмен данными. Два наиболее полезных метода — это **SetText** и **GetText**. Метод **SetText** копирует текст в буфер, замещая все, что до этого было в буфере. Формат использования этого метода следующий:

Clipboard.SetText данные$[,формат%]

Метод **GetText** считывает данные, хранящиеся в буфере. Обращение к этому методу подобно обращению к функции:

данные = Clipboard.GetText()

Используя оба эти метода, нетрудно реализовать такие широко распространенные операции по редактированию текстов, как "Вырезать", "Копировать" и "Вставить".

Еще один полезный метод — **Clear** — позволяет очистить буфер.

Более универсальные методы — **GetData**, **SetData** и **GetFormat** — дают возможность обмениваться не только текстовыми данными. Как уже было упомянуто, буфер позволяет хранить данные различных форматов. Список форматов и соответствующих им констант приведен в следующей таблице.

Имя константы	Значение	Описание
CF_LINK	&HBF00	Динамический обмен данными. Используется для динамической коррекции данных
CF_TEXT	1	Текст
CF_BITMAP	2	Растр
CF_METAFILE	3	Метафайл
CF_DIB	8	Универсальный растр

Последние три формата используются для передачи графических изображений. На рис. 9 приведен пример программы, позволяющей скопировать изображение из одного окна рисунка в другое при помощи буфера хранения.

Метод **GetFormat** применяется для определения соответствия формата данных, хранящихся в буфере, требуемому формату данных. Он бывает нужен тогда, когда существует потенциальная возможность передачи через буфер данных разных форматов (например, между разными программами).

```
'************************************************************
'* Копирование изображений с использованием буфера Clipboard *
'************************************************************
'  В этом примере на форме разместите два окна рисунка
'  и две кнопки (Копировать и Очистить).
'  В первое окно рисунка поместите какой-либо растр,
'  например PARTY.BMP, и скопируйте приведенный текст.

Sub Command1_Click ()
   Clipboard.SetData Picture1.Picture, 2
   Picture2.Picture = Clipboard.GetData(2)
End Sub

Sub Command2_Click ()
   Picture2.Line (0, 0)-Step(Picture2.ScaleWidth,
               Picture2.ScaleHeight), QBColor(15), BF
End Sub
```

Рис. 9

Наибольший интерес представляет динамический обмен данными между программами. В Visual BASIC введен ряд специальных функций и событий, которые значительно упрощают использование DDE. При динамическом обмене между приложениями устанавливаются *горячие связи*, позволяющие немедленно корректировать данные, связанные между собой. Те, кто работал с электронной таблицей Excel, наверняка обратили внимание на то, что при изменении каких-либо данных в таблице мгновенно перестраивается соответствующая диаграмма, изменяются зависимые данные и т.д. Это происходит именно благодаря наличию таких связей. Я могу назвать по крайней мере три приложения, поддерживающих DDE. Это Microsoft Excel, Microsoft Word for Windows и Program Manager. Создавая свои программы на Visual BASIC, вы можете организовать динамическую связь с любой из этих программ, а также динамический обмен данными между вашими прикладными программами.

Два приложения, осуществляющие динамический обмен между собой, выполняют различные функции. Приложение, инициирующее обмен, называется *клиентом*, а приложение, отвечающее ему, — *сервером*. Одно и то же приложение может одновременно осуществлять обмен с несколькими другими приложениями, выступая в одном случае в качестве клиента, а в другом — в качестве сервера. В Visual BASIC любое окно рисунка, окно ввода текста или метка могут быть клиентами, а форма — сервером.

Перед началом динамического обмена необходимо определить две вещи:
- имя приложения, с которым будет осуществляться обмен;
- предмет обмена.

Когда приложение-сервер получает запрос на обмен, касающийся известного ему предмета, оно отвечает на запрос, и обмен начинается. После этого уже невозможно изменить имя приложения или предмет обмена. Сочетание имени и предмета является уникальным для данного обмена и остается постоянным на протяжении всего обмена. Если клиент или сервер изменяют имя или предмет, то обмен незамедлительно прекращается.

Во время обмена клиент и сервер передают друг другу информацию, касающуюся одного или несколь-

ких *элементов обмена.* Элемент обмена представляет собой ссылку на данные, необходимые обоим приложениям. Либо клиент, либо сервер могут сменить элемент обмена без изменения состояния обмена.

Комбинация имени приложения, предмета и элемента обмена однозначно определяет данные, передаваемые между приложениями. Остановимся подробнее на рассмотрении этих трех составляющих.

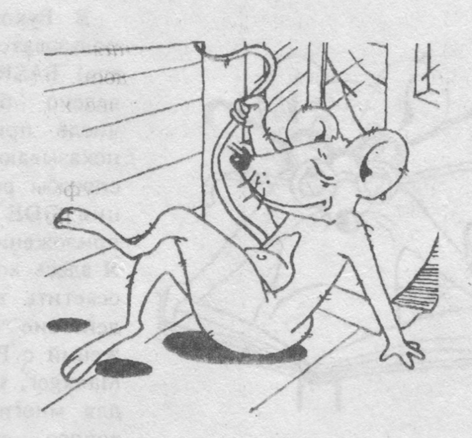
Любое приложение, которое может выступить в качестве сервера при динамическом обмене, имеет свое уникальное имя. Так, уникальное имя Microsoft Excel — "Excel", Microsoft Word for Windows — "WinWord", Program Manager — "PROGMAN". Если в качестве сервера выступает форма, созданная в VB, то ее именем является имя .EXE-файла, а на этапе отладки — имя файла проекта.

В качестве предмета обмена большинство приложений распознают имя документа. Например, для Excel это файлы с расширениями .XLS и .XLC, а для WinWord — с расширениями .DOC. Если в качестве сервера выступает форма VB, то предмет обмена указывается свойством формы LinkTopic. Многие приложения, поддерживающие динамический обмен данными, воспринимают в качестве предмета обмена предмет с названием "System". Данный предмет можно использовать для запроса информации о приложении (поддерживаемые предметы, форматы данных и т.д.).

Элемент обмена идентифицирует часть данных, реально передаваемую в процессе обмена. Например, Excel распознает в качестве элементов обмена ссылки на отдельные ячейки. Если клиентом является какой-либо объект вашей VB-программы, то элемент обмена определяется свойством LinkItem. Если же сервером служит форма, то элементом обмена может быть имя окна ввода текста, метки или окна рисунка. Процесс обмена данными при DDE называют *связью.* Существуют два типа связей, определяющих характер обмена данными между клиентом и сервером:

- горячая связь — сервер передает данные клиенту всякий раз при их изменении;





- холодная связь — сервер передает новые данные клиенту только по его запросу.

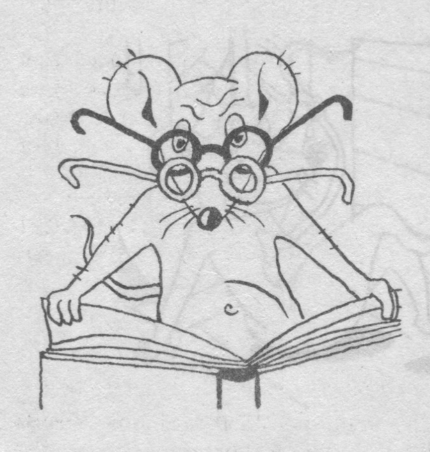
Для того, чтобы установить тот или иной тип связи, используется свойство LinkMode. По умолчанию всем объектам присваивается значение этого свойства None (отсутствует), равное 0. После того как определены имя приложения, предмет и элемент обмена, можно задать тип связи. Для осуществления горячей связи значение LinkMode устанавливается равным 1, для холодной связи — 2.

Динамический обмен данными позволяет дистанционно управлять другими приложениями. Как только клиент установит связь с приложением-сервером, появляется возможность посылать и запрашивать данные, а также посылать команды. Для этого в Visual BASIC имеется четыре метода:
- LinkPoke;
- LinkRequest;
- LinkExecute;
- LinkSend.

Обычно при динамическом обмене данными передача осуществляется от сервера клиенту. Однако метод LinkPoke позволяет передавать данные от клиента серверу.

Как указывалось выше, при установлении горячей связи коррекция данных выполняется автоматически. Если же установлена холодная связь, то автоматическая коррекция не производится. В этом случае для коррекции данных сервер должен запросить их у клиента посредством метода LinkRequest.

Если динамическая связь установлена между текстовыми объектами, то Visual BASIC автоматически корректирует данные. Если же такая связь устанавливается между графическими объектами (окнами рисунка), то автоматическая коррекция не производится. Это связано с тем, что графические растры могут быть довольно большими по объему, и их перерисовка займет значительное время. В этом случае можно использовать метод LinkSend для индикации того, что графический образ изменился и его надо скорректировать.



В Руководстве пользователя Visual BASIC приведено большое число примеров, показывающих способы реализации DDE между приложениями. Я здесь хотел бы осветить взаимодействие приложений с Program Manager, так как для многих этот вопрос не до конца ясен.

В Program Manager динамический обмен данными используется для создания новых групп программ, добавления новых приложений к группам и отображения групп на экране. Для этого в Program Manager имеется ряд специальных функций:

- CreateGroup;
- ShowGroup;
- AddItem;
- DeleteGroup.

```
'****************************************************************
'*  Создание новой группы в Program Manager посредством DDE  *
'****************************************************************
'   На форме разместите метку и кнопку

Function Modify_Group (GrName As String, GrPath As String,
                      FName As String, Description As String)
    On Error Resume Next                'игнорировать ошибки DDE
    'определим программу и ее элемент для динамического обмена
    Label1.LinkTopic = "PROGMAN|PROGMAN"
    Label1.LinkTimeOut = 5
    'установим "холодный" режим DDE
    Label1.LinkMode = 2
    'восстановим на экране Program Manager
    AppActivate "Program Manager"
    SendKeys "%( ){ENTER}", -1
    Z% = DoEvents()
    'создадим новую группу
    Label1.LinkExecute "[CreateGroup(" + GrName + ",
                        " + GrPath + "MyGroup" + ")]"
    'покажем группу
    Label1.LinkExecute "[ShowGroup(" + GrName + ",1)]"
    Z% = DoEvents()
    'добавим новый элемент в группу
    Label1.LinkExecute "[AddItem(" + FName + ",
                          " + Description + "," + FName + ")]"
    Label1.LinkMode = 0
    Z% = DoEvents()
End Function

Sub Command1_Click ()
    A = Modify_Group("New Group", "C:\WIND\",
                "SYSEDIT.EXE", "System Editor")
End Sub

Sub Label1_LinkClose ()
    Label1.LinkMode = 0
End Sub

Sub Form_Load()
    Form.LinkTopic=Label1
End Sub
```

Рис. 10

Для того, чтобы вызвать любую из этих функций, необходимо инициализировать динамический обмен с Program Manager. Перед инициализацией укажите имя приложения "PROGMAN" и предмет обмена "PROGMAN". Далее для непосредственного обращения к функциям Program Manager необходимо использовать метод **LinkExecute**. Формат данного метода зависит от конкретного приложения, но в общем виде он записывается так:

объект.LinkExecute "[команда]"

На выполнение любой команды затрачивается определенное время. Так как при динамическом обмене данными клиент ждет от сервера ответа на сделанный запрос, то очевидно существует время ожидания этого ответа. Если время выполнения команды превысит время ожидания ответа, то система сообщит об ошибке. Программист может изменить это время, меняя значение свойства **LinkTimeOut**. Значение устанавливается в десятых долях секунды и по умолчанию равно 50 (5с). Если это значение установить равным -1, то система не будет сообщать об ошибке и будет ждать ответа до тех пор, пока он не придет. Для ускорения выполнения команды бывает полезно сразу после ее подачи поместить функцию **DoEvents()**.

Теперь кратко о форматах команд Program Manager.

Создание новой группы — функция CreateGroup(GroupName,GroupPath). Здесь GroupName — название группы (использование символов кириллицы не допускается), GroupPath — полный путь и имя файла группы. По умолчанию файл группы помещается в каталог Windows, а имя файла группы образуется из первых 8 символов имени группы.

Отображение группы — функция ShowGroup(GroupName,ShowCommand). GroupName — необязательный параметр. Если он опущен, то команда относится к группе, выбранной в данный момент. ShowCommand — команда, сообщающая Диспетчеру программ, как именно следует отобразить группу (размер по умолчанию + фокус, полноэкранно, в виде иконки и т.п.).

Добавление нового приложения в группу — функция AddItem(CmdLine,[Name,[IconPath,IconIndex,[xPos,yPos]]]). Параметр CmdLine содержит полный путь поиска данного приложения. Необязательный параметр Name — название приложения. Оно появляется под иконкой, соответствующей приложению. IconPath — необязательный параметр, указывающий путь к файлу иконки (если используется иконка, не включенная в .EXE-файл). IconIndex — также необязательный параметр. Указывается в том случае, если используется иконка, расположенная в теле PROGMAN.EXE. В Windows 3.0 Диспетчер Программ имеет 6 встроенных иконок. Параметры xPos, yPos — необязательные параметры, определяющие положение иконки относительно окна группы. Если они не указаны, то размещение происходит автоматически.

Функция DeleteGroup(GroupName) применяется для удаления группы.



Пример программы, использующей данные функции, приведен на рис. 10.

В конце обзора способов обмена данными между приложениями в среде Windows я хотел бы кратко остановиться на связи и внедрении объектов (OLE).

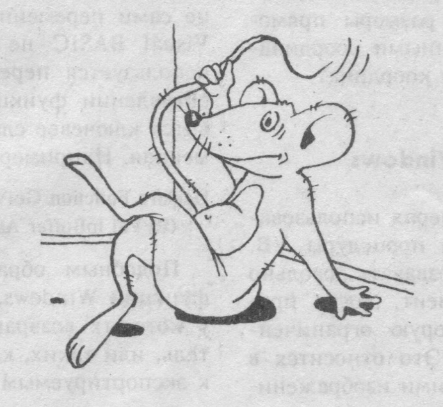
Поясним суть данного способа на следующем примере. Допустим, вы работаете с текстовым процессором и редактируете некоторый текст. Этот текст содержит графику и какие-нибудь другие нетекстовые объекты. Подведя курсор мыши к графическому объекту и дважды щелкнув кнопкой, вы вызовете ИМЕННО ТУ программу, в которой был создан данный объект. Изменив этот объект в этой программе, вы возвращаетесь в текстовый процессор и обнаруживаете, что там графический образ тоже изменился. Такое поведение чем-то напоминает протокол DDE, однако здесь есть существенное отличие. Элементами обмена тут являются крупные структуры-объекты, изменение которых возможно только в той программе, которая их внедрила.

На момент написания данной статьи протокол связи и внедрения объектов используют такие программы, как MS Excel 3.0, MS Works for Windows, MS Publisher и MS Word for Windows 2.0. Пакет Visual BASIC 1.0 не имеет встроенных функций поддержки OLE, поэтому реализация этого способа обмена данными в настоящей статье не рассматривается.

Использование таймера

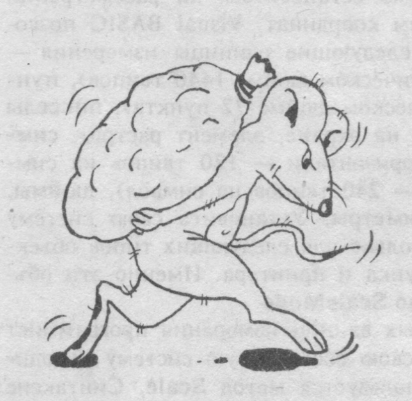
В Windows таймер играет важную роль. По его сигналам (в применении к DOS это звучало бы "по его прерываниям") можно не только определять дату или время, но и контролировать текущее состояние окна программы, передавать управление другим программам или процедурам, выполнять какие-либо фоновые операции.

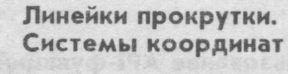
В Visual BASIC таймер реализован в виде отдельного объекта управления, который, подобно другим объектам, можно разместить на форме. Размер и местоположение таймера не имеют значения, так как он отображается на экране только на этапе работы с проектом. Во время выполнения программы он не виден. Пользователь может запрограммировать таймер так, как ему будет нужно. Основное свойство таймера — Interval (интервал) — определяет в миллисекундах продолжительность паузы между событиями. Значение интервала может изменяться от 0 до 64767; то есть самая длинная пауза лишь слегка превышает минуту. Заметим, что в действительности длительность интервала может отличаться от определяемого им точного временного промежутка, поэтому при необходимости следует обратиться к функции Time$ для коррекции времени. Несмотря на то, что единицей интервала является миллисекунда, фактически он задается с точностью до 1/18 секунды, так как прерывание от таймера генерируется в системе только 18 раз в секунду.



По истечении заданного интервала времени таймер генерирует событие Timer. Программист может использовать наступление этого события в своих целях.

В Windows существует ограничение на число одновременно работающих таймеров — оно не должно превышать 16. Причем сюда входят все таймеры системы. Чтобы в этом убедиться, откройте несколько окон с часами или календарем. Как только максимально допустимое число будет превышено, Windows сообщит о невозможности открытия большего количества таймеров.





В качестве примера использования таймера на рис. 11 приведен текст программы-часов. В своей работе данная программа подобна часам, входящим в комплект Windows, но имеет чисто внешние отличия.

Линейки прокрутки. Системы координат

Любой, кто работал в среде Windows, наверняка обратил внимание на *линейки прокрутки*, расположенные обычно в правой или нижней части окна. Использование линеек прокрутки значительно облегчает работу с текстом или с графикой больших

размеров. Линейка прокрутки имеет четыре активных области — верхняя (левая) и нижняя (правая) кнопки, бегунок и тело линейки. Нажатие на кнопки приводит к небольшой прокрутке изображения, в то время как нажатие на тело линейки приводит к значительно большей прокрутке. Кроме того, изображение можно прокрутить в окне, если "потянуть" непосредственно за бегунок.

В Visual BASIC введены два объекта — вертикальная и горизонтальная линейки прокрутки. Эти объекты имеют следующие свойства: Max, Min, SmallChange, LargeChange, Value — соответственно максимальное и минимальное значения, малое и большое приращения и текущее значение. Минимальное и максимальное значения могут быть установлены в пределах от 0 до 32767. По умолчанию малое и большое приращения устанавливаются равными 1. Программист в своей программе определяет эти значения в соответствии со своими потребностями. Пример использования линейки прокрутки можно найти в тексте программы, приведенной на рис. 12.

Теперь необходимо остановиться на рассмотрении применяемых систем координат. Visual BASIC позволяет использовать следующие единицы измерения — **твипы** (в одном логическом дюйме 1440 твипов), **пункты** (в одном логическом дюйме 72 пункта), **пикселы** (мельчайшая точка на экране, элемент растра), **символы** (размер по горизонтали — 120 твипов на символ, по вертикали — 240 твипов на символ), **дюймы**, **миллиметры**, **сантиметры**. Установить свою систему координат можно только для следующих типов объектов — формы, рисунка и принтера. Именно эти объекты имеют свойство **ScaleMode**.

Помимо указанных единиц измерения программист может определить свою собственную систему координат. Для этого используется метод **Scale**. Синтаксис определения новой системы координат следующий:

```
[объект].Scale (x1!,y1!)-(x2!,y2!),
```

где x1 и y1 — координаты левого верхнего угла объекта, а x2 и y2 — правого нижнего. Изменение системы координат влияет на положение графики или объектов. Так, одной лишь сменой системы координат можно переместить графику или изменить ее размеры. На рис. 12 приведен пример простой программы, позволяющей перемещать и изменять размеры прямоугольника, определяемого фиксированными координатами, посредством изменения системы координат.

Использование API-функций Windows

Во всех рассмотренных ранее примерах использовались только встроенные функции и процедуры VB. Несмотря на то, что они позволяют создавать довольно сложные программы, наступает момент, когда программист начинает понимать некоторую ограниченность возможностей этих функций. Это относится в первую очередь к работе с графическими изображениями, работе с последовательными портами компьютера, работе с инициализирующими файлами и т.п.

К характерным особенностям Visual BASIC относится простое расширение его возможностей за счет использования стандартных функций Windows. Те, у кого уже есть опыт программирования для Windows, знают, что в среде Windows имеется порядка 500 собственных функций. Эти функции содержатся в файлах USER.EXE, KERNEL.EXE и GDI.EXE. Способы обращения к этим функциям и их типы подробно описаны в Руководстве программиста SDK.

Для того, чтобы использовать какую-либо из функций API Windows в программе на VB, нужно предварительно объявить ее. С этой целью необходимо в глобальном модуле программы ввести оператор **Declare**. Формат его следующий:

```
Declare Sub «имя процедуры» Lib «имя библиотеки»
        [Alias «реальное имя»] (параметры)
    или
Declare Function «имя функции» Lib «имя библиотеки»
        [Alias «реальное имя»] (параметры) As «тип функции»,
```

где имя процедуры (имя функции) — то имя, которое будет использовано в вашей программе; имя библиотеки — название файла, в котором находится данная функция (для стандартных функций это "User", "Kernel", "GDI"); Alias «реальное имя» — опционный параметр, используется в случае, если реальное имя функции содержит символы, недопустимые в Visual BASIC (так, если имя функции начинается с подчерка, то следует использовать эту конструкцию псевдонима).

Необходимо кратко остановиться на списке параметров. В нем через запятую указываются параметры, передаваемые функции и их типы. Некоторые функции допускают передачу одного и того же параметра разных типов. Чтобы отключить в VB проверку на соответствие типов, устанавливается тип параметра **As Any**.

Помимо этого, очень часто при обращении к стандартным функциям Windows возникает потребность передачи параметра *по значению*. Поясню это. Большинству функций в качестве параметров передаются не сами переменные, а указатели на них. Так как в Visual BASIC не существует понятия указатель, то используется передача **по значению**. Для этого при объявлении функции перед именем переменной ставится ключевое слово **ByVal**, затем следует сама переменная. Например:

```
Declare Function GetWindowsDirectory Lib "Kernel"
    (ByVal lpBuffer As String, ByVal nSize As Integer) As Integer
```

Подобным образом можно объявить большинство функций Windows, за исключением некоторых — тех, у которых возвращаемым значением является указатель, или таких, которые обращаются непосредственно к экспортируемым функциям основной программы.

```
'*****************************************************************
*                                                                *
'* Стрелочные и цифровые часы. Демонстрационная программа.      *
'* (c) Ф.Зубанов, 1991                                           *
'*****************************************************************
' На форме разместите окно рисунка (CtlName="clock") и таймер,
' а также создайте меню:
'       Справка           IdHelp
'       О программе       IdAbout
'       Вид часов         IdType
'       Стрелочные        IdNormal
'       Цифровые          IdDigit
Dim Radius As Integer             'радиус циферблата
Dim CurrentHour As Integer        'текущее значение часов
Dim CurrentMinute As Integer      'текущее значение минут
Dim PrevMinute As Integer         'предыдущее значение минут
Dim CurrentSecund As Integer      'текущее значение секунд
Dim Title As String               'заголовок программы
Dim First As Integer              'логическая переменная
'эти переменные используются для изображения стрелок
Dim x0 As Integer
Dim hx0 As Integer
Dim mx0 As Integer
Dim y0 As Integer
Dim hy0 As Integer
Dim my0 As Integer
'координаты центра циферблата
Dim CPX As Integer
Dim CPY As Integer
Sub Timer1_Timer ()               'пришло событие от таймера
  If IdNormal.Checked Then
    ShowNormalClock               'если в меню выбраны стрелочные часы
  Else
    ShowDigitClock                'если в меню выбраны цифровые часы
  End If
End Sub

Sub Form_Resize ()                'при изменении размеров формы надо
  DrawFace                        'перерисовать часы
End Sub

Sub Form_Load ()                  'инициализация переменных при старте
  First = -1
  PrevHour = -1
  PrevMinute = -1
  Title = "Часы"
  'в заголовок программы включим текущую дату
  Form1.Caption = Title + " - " + Format$(Now, "dd-mmm-yyyy")
End Sub

Sub ShowNormalClock ()            'данная процедура рисует стрелки
 Pi# = 4 * Atn(1)
 CurrentHour = Val(Time$)
 CurrentMinute = Val(Mid$(Time$, 4, 2))
 CurrentSecund = Val(Mid$(Time$, 7))
 'вычислим координаты конца секундной стрелки
 x% = Radius * Cos((CurrentSecund - 15) * 6 * Pi# / 180)
 y% = Radius * Sin((CurrentSecund - 15) * 6 * Pi# / 180)
 'вычислим координаты конца минутной стрелки
 mx% = Radius * Cos((CurrentMinute - 15) * 6 * Pi# / 180)
 my% = Radius * Sin((CurrentMinute - 15) * 6 * Pi# / 180)
 'вычислим координаты конца часовой стрелки
 '(меняется каждые 12 минут)
 hx% = (Radius - 240)*Cos(((CurrentHour)*5 + CurrentMinute \
        12 - 15) * 6 * Pi# / 180)
 hy% = (Radius - 240)*Sin(((CurrentHour)*5 + CurrentMinute \
        12 - 15) * 6 * Pi# / 180)
 'установим XOR вывод изображения
 clock.DrawMode = 7
 If First Then            'если первичный вывод изображения, то
   clock.DrawMode = 13
'стрелки рисуются только один раз
     DrawFace
     clock.DrawMode = 7
     First = 0
     clock.Line (CPX, CPY)-Step(x%, y%), QBColor(4)
     clock.DrawWidth = 3
     clock.Line (CPX, CPY)-Step(hx%, hy%), QBColor(14)
     clock.Line (CPX, CPY)-Step(mx%, my%), QBColor(14)
     clock.DrawWidth = 1
     x0 = x%
     y0 = y%
     hx0 = hx%
     hy0 = hy%
     mx0 = mx%
     my0 = my%
 Else                     'в противном случае надо стереть
                          'стрелки в предыдущем положении
     clock.Line (CPX, CPY)-Step(x0, y0), QBColor(4)
     clock.Line (CPX, CPY)-Step(x%, y%), QBColor(4)
     x0 = x%
     y0 = y%
     clock.DrawWidth = 3
     If (PrevMinute \ 12) <> (CurrentMinute \ 12) Then
       clock.Line (CPX, CPY)-Step(hx0, hy0), QBColor(14)
       clock.Line (CPX, CPY)-Step(hx%, hy%), QBColor(14)
       hx0 = hx%
       hy0 = hy%
       PrevHour = CurrentHour
     End If
     If PrevMinute <> CurrentMinute Then
       clock.Line (CPX, CPY)-Step(mx0, my0), QBColor(14)
       clock.Line (CPX, CPY)-Step(mx%, my%), QBColor(14)
       mx0 = mx%
       my0 = my%
       PrevMinute = CurrentMinute
     End If
     clock.DrawWidth = 1
   End If
   clock.DrawMode = 13   'восстановим нормальный режим
End Sub

Sub IdNormal_Click ()          'если в меню выбраны стрелочные часы
   IdNormal.Checked = -1
   IdDigit.Checked = 0
   First = -1
   clock.Cls
End Sub
Sub IdDigit_Click ()           'если в меню выбраны цифровые часы
   IdNormal.Checked = 0
   IdDigit.Checked = -1
   DrawFace
End Sub
Sub ShowDigitClock ()
   clock.Cls
   clock.Print Time$
End Sub
Sub DrawFace ()           'данная процедура перерисовывает циферблат
   clock.width = Form1.ScaleWidth
   clock.Height = Form1.ScaleHeight
   CPX = clock.width \ 2
   CPY = clock.Height \ 2
   If clock.Height < clock.width Then
     Radius = clock.Height \ 2 - 140
   Else
     Radius = clock.width \ 2 - 140
   End If
   First = -1
   clock.Cls
   If IdNormal.Checked Then
   Pi# = Atn(1) * 4
   clock.fontsize = 12
   clock.DrawWidth = 2
   clock.Circle (CPX, CPY), Radius, RGB(&HFF, 0, 0)
   clock.DrawWidth = 1
   For I% = 1 To 12
   clock.CurrentX = CPX + (Radius - 240) * Cos((I% - 3) * 30 *
                  Pi# / 180) - clock.TextWidth(Str$(I%)) \ 2
   clock.CurrentY = CPY + (Radius - 240) * Sin((I% - 3) * 30 *
                  Pi# / 180) - clock.TextHeight(Str$(I%)) \ 2
   clock.Print Str$(I%);
   Next
   Else
   clock.fontsize = clock.width \ 100
   End If
End Sub
Sub IdAbout_Click ()      'если в меню выбран пункт "О программе"
   NL$ = Chr$(13) + Chr$(10)
   Msg$ = Chr$(9) + "Комбинированные часы" + NL$
   Msg$ = Msg$+"Реализованы на Microsoft Visual BASIC 1.0"+NL$
   Msg$ = Msg$ + Chr$(9) + "(c) Ф.Зубанов 1991"
   MsgBox Msg$, 0, "Часы"
End Sub
```

Рис. 11

После того как функция объявлена, ее можно использовать в любой части программы так же, как и стандартную функцию Visual BASIC. Прототипы функций описаны в файле WINDOWS.H (входит в SDK или QCWin). Зная прототип функции, нетрудно объявить ее в VB. Однако имеется специальный файл WINAPI.TXT, содержащий объявления всех доступных функций, а также объявления типов переменных и констант Windows. Этот файл можно получить либо в Compuserve (область MSBASIC, Dl5), либо в московском представительстве фирмы Microsoft.

В качестве примера использования API-функций приведен текст программы на рис. 13. Данная программа определяет конфигурацию вашего компьютера, режимы работы Windows, а также ключевые каталоги. Текст программы снабжен комментариями и не требует дополнительных пояснений.

Использование функций, написанных на других языках программирования

В предыдущем разделе было показано, как в Visual BASIC можно использовать стандартные функции API-Windows. Файлы, в которых находятся эти функции, являются динамическими библиотеками Windows. Естественно задать вопрос, а нельзя ли таким же образом использовать любые функции, размещаемые в динамических библиотеках? Конечно можно! Но при написании таких функций нужно соблюдать соглашения о вызовах, принятые в языке Pascal, — число передаваемых параметров должно быть фиксированным, передача параметров в стек идет слева направо, и функция сама восстанавливает стек по завершении работы. Необходимо также помнить о том, что Visual BASIC не может оперировать с указателями, поэтому значения, возвращаемые функцией, не должны быть указателями.

Данное свойство VB открывает широкий простор для программистов, желающих использовать свои собственные нестандартные решения в программах на VB. В качестве примера я хотел бы привести процедуру, выполняющую копирование файлов из одного каталога в другой с одновременной распаковкой файлов. Как известно, все файлы на инсталляционных дискетах Windows и ряда других программ хранятся в сжатом виде. Программа, выполняющая это сжатие, поставляется в составе SDK и называется Compress. Для обратного преобразования файлов можно либо использовать программу Expand (DOS), либо воспользоваться функциями динамической библиотеки LZEXPAND.DLL, находящейся в каталоге Windows.

К сожалению, функции, входящие в эту динамическую библиотеку, не документированы, поэтому последующее описание можно считать верным только для версии Windows 3.0. Для работы со сжатыми файлами используются функции lzOpenFile, lzSeek, lzRead, lzClose. При выполнении операции копирования необходимо оставлять дату создания файла неизменной. Для этого нужно написать еще одну функцию, выполняющую такую операцию. На рис. 14 приведен пример функции копирования. Туда же включен текст функции, восстанавливающей дату создания файла. Эта функция написана на языке С и может быть скомпилирована любым компилятором С.

```
' Расположите на форме окно рисунка, горизонтальную
' линейку прокрутки, две отметки выбора, названные
' "сдвиг" и "масштабирование"

Const TRUE = -1
Const FALSE = 0

Dim X0 As Integer
Dim X1 As Integer

Sub Form_Load ()                             'инициализация переменных.
  X0 = 0
  X1 = 1000
  Picture1.ScaleMode = 3                     'единицы измерения -- пикселы
  Picture1.Scale (X0,0)-(X1,1000)            'установка начальной шкалы
  HScroll1.Max = 1000                        'максимальное значение -- 5000
  HScroll1.LargeChange = 2000                'большое приращение
  HScroll1.SmallChange = 20                  'малое приращение
End Sub

Sub HScroll1_Change ()                       'изменение положения бегунка
  'если выбрана опция сдвига, то изменять координаты синхронно
  'и в одну сторону, иначе изменять координаты
  'в противоположные стороны
  If Option1.Value Then
     Picture1.Scale (X0 + HScroll1.Value, 0)-
                    (X1 + HScroll1.Value, 1000)
  Else
     Picture1.Scale (X0 - HScroll1.Value, 0)-
                    (X1 + HScroll1.Value, 1000)
  End If
  Picture1.Cls                               'очистить изображение
  'и нарисовать прямоугольник с фиксированными координатами
  Picture1.Line (250, 250)-(750, 750), QBColor(4), BF
End Sub

Sub Option1_Click ()                         'выбираем перемещение
  Option1.Value = TRUE
  Option2.Value = FALSE
End Sub

Sub Option2_Click ()                         'выбираем масштабирование
  Option2.Value = TRUE
  Option1.Value = FALSE
End Sub

Sub Form_Paint ()
  Picture1.Line (250, 250)-(750, 750), QBColor(4), BF
End Sub
```

Рис. 12

Из сказанного выше понятно, что функцию можно написать на любом языке программирования, позволяющем создавать динамические библиотеки для Windows и использующем соглашение о вызовах функций, принятое в Pascal. К ним относятся Microsoft С и Quick С for Windows, Borland C++, Turbo Pascal for Windows, Zortech C++, Microsoft Fortran 5.1 и ряд других компиляторов. А отсюда следует вывод о том, что практически любую программу, написанную для работы в DOS, легко перевести в программу для Windows. На мой взгляд, особый интерес это может вызвать у программистов на FORTRAN. FORTRAN характеризуется значительной вычислительной мощью и сравнительной неразвитостью графического оформле-

```
'Для данной программы разместите на форме 8 меток:
'       "Тип процессора"
'       "Сопроцессор"
'       "Режим работы Windows"
'       "Защищенный режим"
'       "EMS"
'       "Каталог Windows"
'       "Каталог System"
'       "Текущий каталог"
'и напротив каждой из них объект "Picture".
'Внизу формы разместите кнопку "OK".

'В глобальном модуле объявите:
Declare Function GetWinFlags Lib "Kernel" () As Long
Declare Function GetWindowsDirectory Lib "Kernel"
    (ByVal lpBuffer As String,ByVal nSize As Integer) As Integer
Declare Function GetSystemDirectory Lib "Kernel" (ByVal lpBuffer
                         As String, ByVal nSize As Integer) As Integer

    Global Const WF_PMODE = &H1
    Global Const WF_CPU286 = &H2
    Global Const WF_CPU386 = &H4
    Global Const WF_CPU486 = &H8
    Global Const WF_STANDARD = &H10
    Global Const WF_ENHANCED = &H20
    Global Const WF_CPU086 = &H40
    Global Const WF_CPU186 = &H80
    Global Const WF_LARGEFRAME = &H100
    Global Const WF_SMALLFRAME = &H200
    Global Const WF_80x87 = &H400

'А это введите в файле формы
Dim WindowsDir As String
Dim SystemDir As String

Sub Form_Paint ()
    A& = GetWinFlags()
    'работает в защищенном режиме?
    If A& And WF_PMODE Then
        Picture4.Print "Да"
    Else
        Picture4.Print "Нет"
    End If
    'определим тип процессора
    If A& And WF_CPU286 Then
        Picture1.Print "Intel286"
    ElseIf A& And WF_CPU386 Then
        Picture1.Print "Intel386"
    ElseIf A& And WF_CPU486 Then
        Picture1.Print "Intel486"
    ElseIf A& And WF_CPU086 Then
        Picture1.Print "Intel8086/8088"
    ElseIf A& And WF_CPU186 Then
        Picture1.Print "Intel186"
    Else
        Picture1.Print "неизвестный"
    End If
    'определим режим работы Windows
    If A& And WF_STANDARD Then
        Picture3.Print "Стандартный"
    ElseIf A& And WF_ENHANCED Then
        Picture3.Print "Расширенный"
    Else
        Picture3.Print "Реальный"
    End If
    'тип фрейма EMS
    If A& And WF_LARGEFRAME Then
        Picture5.Print "Большой фрейм"
    ElseIf A& And WF_SMALLFRAME Then
        Picture5.Print "Малый фрейм"
    Else
        Picture5.Print "отсутствует"
    End If
    'определим наличие сопроцессора
    If A& And WF_80x87 Then
        Picture2.Print "присутствует"
    Else
        Picture2.Print "отсутствует"
    End If
    WindowsDir = String$(128, 32)
    SystemDir = String$(128, 32)
    If GetWindowsDirectory(WindowsDir, 128) Then
        Picture6.Print RTrim$(WindowsDir)
    Else
        Picture6.Print "Не определен"
    End If
    If GetSystemDirectory(SystemDir, 128) Then
        Picture7.Print RTrim$(SystemDir)
    Else
        Picture7.Print "Не определен"
    End If
    Picture8.Print CurDir$
End Sub

Sub Command1_Click ()
    End
End Sub
```

Рис. 13

ния программ. В этом случае Visual BASIC удобно использовать для создания внешнего интерфейса с пользователем, а все функции, написанные на FORTRAN, переместить в динамическую библиотеку и объявить в глобальной части программы на VB.

Использование новых объектов управления

Как было упомянуто в начале статьи, в Visual BASIC нельзя создавать новые объекты управления средствами самого VB. Программист вынужден пользоваться тем набором, который предлагается в основной поставке. Однако существует огромный класс объектов, использование которых интересно для широкого круга пользователей. Это, например, такие объекты, как электронная таблица, синтезатор звука, кнопки с несколькими устойчивыми состояниями и т.п.

Все эти объекты можно создать посредством еще одного продукта фирмы Microsoft — Visual BASIC Control Development Kit (VB CDK). В его комплект входят библиотеки, позволяющие создавать новые объекты управления. Язык программирования — C. В комплект включены также тексты некоторых новых объектов управления и подробная документация.

Созданный новый объект по существу является динамической библиотекой Windows. Однако расширение у файлов объектов иное — .VBX. Для использования нового объекта необходимо в меню Visual BASIC выбрать **File Add** и загрузить один или несколько новых объектов. Как только объект загружен, в окне объектов появляется новая иконка, соответствующая данному объекту. Дальнейшая работа с новыми объектами ничуть не отличается от работы с предопределенными объектами. Однако надо помнить, что если вы комулибо передаете свою готовую программу, то вместе с

```
'***************************************************************
'*    Пример использования функций, написанных на других      *
'*                языках программирования.                     *
'*    Копирование с одновременной распаковкой файлов           *
'***************************************************************

'эти строки надо скопировать в глобальный модуль
'  OpenFile() Structure
Type OFSTRUCT
    cBytes As String * 1
    fFixedDisk As String * 1
    nErrCode As Integer
    reserved As String * 4
    szPathName As String * 128
End Type

'  OpenFile() Flags
Global Const OF_READ = &H0
Global Const OF_WRITE = &H1
Global Const OF_READWRITE = &H2

Declare Function lzOpenFile Lib "lzexpand.dll"
       (ByVal lpFileName As String, lpReOpenBuff As OFSTRUCT,
        ByVal wStyle As Integer) As Integer
Declare Function lzSeek Lib "lzexpand.dll"
       (ByVal hFile As Integer, ByVal lOffset As Long,
        ByVal iOrigin As Integer) As Long
Declare Function lzRead Lib "lzexpand.dll"
       (ByVal hFile As Integer, ByVal lpBuffer As String,
        ByVal wBytes As Integer) As Integer
Declare Sub lzClose Lib "lzexpand.dll" (ByVal hFile As Integer)
Declare Function UpdateFileDate Lib "free.dll"
    (ByVal Sourse As String, ByVal Dest As String) As Integer

Global Const BLOCK = 32000   'копирование больших файлов выпол-
няется блоками
Global Const TRUE = -1
Global Const FALSE = 0

'последующие строки копируются в файл формы

Function ExpandCopy_File
          (Source As String, Dest As String) As Integer
   On Error GoTo errhandler
   hSource = lzOpenFile(Source, OFSTR, OF_READ)
   FileFree = FreeFile
   NUL& = CLng(0)
   Open Dest For Binary Access Write As FileFree
   N& = lzSeek(hSource, NUL&, 2)
   L& = lzSeek(hSource, NUL&, 0)
   NLD& = N& - L&
   Do Until L& >= NLD&
        M& = L&
        Temp$ = String$(BLOCK, 32)
        P& = lzRead(hSource, Temp$, BLOCK)
        L& = L& + P&
        Temp$ = Left$(Temp$, P&)
        Put FileFree, , Temp$
        Dell% = DoEvents()
   Loop
   lzClose (hSource)
   Close (FileFree)
   Z% = UpdateFileDate(Source, Dest)
        ExpandCopy_File = TRUE
        Exit Function
errhandler:
        ExpandCopy_File = FALSE
        ErrMes$="Error during writing file"
        MsgBox ErrMes$
        Resume
        Exit Function
End Function
```

Далее следует текст на С для создания файла FREE.DLL

```
//*************************************************************
//*   This is DLL supporting VB function ExpandCopyFile       *
//*   (C) Zubanov F. 1991                                     *
//*************************************************************
// эта dll скомпилирована в MS QuickC for Windows 1.0
// при компиляции другими компиляторами добавьте функцию WEP

 #include <windows.h>
 #include <dos.h>
 WORD  wDate, wTime;

int FAR PASCAL LibMain (HANDLE hModule, WORD wDataSeg,
                  WORD wHeapSize, LPSTR lpszCmdLine) {
  // Сегмент данных может перемещаться
  if (wHeapSize != 0) UnlockData(0);
    return(1); // возвратить TRUE, если инициализировано удачно
}

int FAR PASCAL UpdateFileDate  (LPSTR szSrcPath,
                              LPSTR szDstPath){

  int nSrcFile, nDstFile;
  OFSTRUCT ofSrc, ofDst;

  nSrcFile = OpenFile(szSrcPath, &ofSrc, OF_READ);
  if (nSrcFile == -1)
    return 0;

  nDstFile = OpenFile(szDstPath, &ofDst, OF_WRITE);
  if (nDstFile == -1)
    return 0;

  _dos_getftime(nSrcFile, &wDate, &wTime);
  _dos_setftime(nDstFile, wDate, wTime);
  _lclose(nDstFile);
  _lclose(nSrcFile);
  return -1 ;
}
```

А это текст файла FREE.DEF

```
*************************************************************
; Module name: Free.DEF
; Programmer : Zubanov Fyodor

LIBRARY      FREE
DESCRIPTION  'Sample dll for VB support'
EXETYPE      WINDOWS
STUB         'WinStub.Exe'
CODE         PRELOAD  MOVEABLE DISCARDABLE
DATA         PRELOAD  MOVEABLE SINGLE
HEAPSIZE     1024
EXPORTS
UpdateFileDate @1
```

Рис. 14

.EXE-файлом необходимо передать и все .VBX файлы, используемые в вашей программе.

К сожалению, рассказ о создании новых объектов выходит за рамки данной статьи. Единственное, что хотелось бы добавить — это то, что ряд фирм уже поставляет дополнительные объекты для VB, перекрывающие практически все потребности программистов. Так, например, фирма MicroHelp разработала очень большой набор объектов, включающих новые модифицированные кнопки, списки с возможностью множественного выбора, музыкальные объекты, индикаторы и многое другое. В Compuserve можно найти примеры демонстрационных программ для продуктов этой фирмы.

Заключение

Охватить все тонкости программирования на Visual BASIC не может ни одна журнальная статья. Да я и

не ставил перед собой такой цели. Главное — это дать почувствовать идеологию программирования на Microsoft Visual BASIC, показать основные приемы и ответить на некоторые вопросы, встречающиеся на пути программистов. Большую помощь в этом мне оказали дискуссии с зарубежными коллегами-программистами в сети Internet (область comp.windows.ms.programmer), наблюдение за дискуссиями в Fido-Net, материалы, помещенные в Compuserve в области MSBASIC, а также собственный опыт программирования на VB. Надеюсь, что по мере накопления новых материалов у меня будет возможность их публикации.

В конце статьи приведен краткий словарь терминов, используемых в Visual BASIC и в программировании для Windows, с пояснениями их значений.

Ф.Зубанов

Использованы материалы:

1. Microsoft Visual BASIC Programmers Reference.
2. Microsoft Visual BASIC Language Reference.
3. Microsoft Windows Software Developement Kit, Reference, v.1.
4. Microsoft Windows Software Developement Kit, Guide to Programming.

Список использованных терминов

Русский термин	Английский термин	Описание
буфер хранения	clipboard	объект VB, служащий для передачи информации между объектами управления
горячая связь	hot link	способ динамического обмена данными, при котором после внесения каких-либо изменений немедленно осуществляется коррекция информации
динамический обмен данными	dinamic data exchange (DDE)	способ обмена данными между объектами или программами, подразумевающий наличие горячих связей между ними
заголовок	caption	заголовок объекта управления, выводимый на экран
иконка	icon	небольшая картинка, являющаяся графическим образом программы и изображаемая вместо окна программы при ее минимизации
клиент	client	объект, инициализирующий динамический обмен данными с сервером
кнопка	command button	объект управления VB, имитирующий кнопку
комбинированный список	combo box	объект управления VB, объединяющий окно ввода текста и список
линейка прокрутки	scroll bar	полоса, расположенная в правой или нижней части окна и позволяющая производить прокрутку содержимого окна
метка	label	объект управления VB, позволяющий помечать отдельные части формы, а также служащий для осуществления динамической связи
объект	object	глобальная структура VB, определяющая свойства объектов управления
объект управления	control	основная структурная единица программы на Visual BASIC
окно рисунка	picture box	объект управления VB, позволяющий выводить изображения и текст
окно ввода текста	text box	объект управления VB, позволяющий вводить и редактировать текст
отладчик	debug	объект VB, позволяющий выводить промежуточные данные в окно Immediate
отметка выбора	option button	объект управления VB, указывающий на выбор одной из опций
предмет обмена	topic	параметр, который совместно с элементом обмена однозначно определяет информацию при динамическом обмене данными
принтер	printer	объект VB, определяющий свойства объектов управления относительно принтера
рамка	frame	объект управления VB, служащий для визуального объединения группы объектов управления на экране
связь и внедрение объектов	object linking and embedding (OLE)	способ обмена информацией между программами
сервер	server	объект, участвующий в динамическом обмене данными и отвечающий на запрос клиента
список	list box	объект управления VB, позволяющий выводить на экран какой-либо список, осуществлять выбор из этого списка и сортировать по алфавиту
список дисков	drive list box	комбинированный список, в котором перечислены доступные для данного компьютера диски
список каталогов	directory list box	комбинированный список, в котором перечислены в алфавитном порядке все каталоги выбранного диска
список файлов	file list box	комбинированный список, в котором перечислены в алфавитном порядке все файлы выбранного каталога
таймер	timer	объект управления VB, служащий для генерации временных интервалов
флажок	check box	объект управления VB, указывающий на выбор опции
форма	form	основной объект управления VB, служащий для размещения на нем остальных объектов
холодная связь	cold link	способ динамического обмена данными, при котором коррекция информации осуществляется только по запросу сервера
экран	screen	объект VB, определяющий все свойства объектов управления относительно экрана дисплея
элемент обмена	item	параметр, который совместно с предметом обмена однозначно определяет информацию при динамическом обмене данными

«С-Сервис»

МОЩНАЯ ИНСТРУМЕНТАЛЬНАЯ ОБОЛОЧКА
ДЛЯ ПРОГРАММИСТОВ НА ЯЗЫКЕ СИ

Самый высокий уровень сервиса !

Работа без файлов — только с объектами и функциями, все остальное оболочка сделает за Вас !
Мгновенная фиксация синтаксических ошибок до компиляции !

Новая разработка предприятия "Семигор" — такого сервиса вы не найдете в оболочках Microsoft, Borland, Zortech.

Инструментальная оболочка включает:

- ☐ уникальное средство создания, ведения и модификации деревьев проекта;
- ☐ настраиваемый многооконный редактор;
- ☐ средства фиксации языковых ошибок непосредственно в месте и в момент возникновения (до компиляции !!!) на основе "непрерывного" и "мгновеннного" лексического и синтаксического анализа;
- ☐ средства визуальной идентификации логической структуры проекта (построение дерева иерархии вызовов функций) — полная ясность замысла и его воплощения;
- ☐ средства автоматической диспетчеризации файлов, предъявляемых к компиляции (в которых размещаются функции, вводимые пользователем), по эффективным стратегиям — забудьте про файлы и Вы попадете в страну объектов и функций, в которой так легко дышится настоящему знатоку С;
- ☐ средства ведения разнообразных таблиц: функций проекта с их описанием; глобальных переменных с указанием их типов и функций, на которые данные объявления распространяются; констант и типов переменных — Вы будете знать все !
- ☐ средства визуального отслеживания правильности создаваемых языковых конструкций по дереву синтаксиса языка С — заботливый МЕНТОР всегда с Вами !

Оболочка "С-Сервис" отвечает логике разработчика: Алгоритм (спецификации) — Программа (функции) — Документация. При этом на каждом этапе Вам гарантируются ясный обзор всего проекта, соответствие архитектуре и большой сервис.

В следующей версии в состав инструментальной оболочки будет включен оригинальный компилятор Tree Compiler, обладающий сверхвысоким быстродействием (время компиляции — несколько секунд при любом объеме листинга!).

Техническая документация на русском языке. Поставляются бесплатные инсталляции. Гарантия 1 год.

В следующей версии — сверхбыстрая компиляция ! ! !

440000,
г. Пенза, а/я 72

«СЕМИГОР»

Телефакс: (841-2) 64-78-50
Телетайп: 155349 РИФ



Прошло более десяти лет с момента появления первой электронной таблицы для персональных компьютеров — VisiCalc. VisiCalc имела невероятный успех и привлекла внимание бизнесменов к персональным компьютерам. Электронная таблица фирмы Lotus Developtent сыграла такую же роль при появлении IBM PC. После этого появилось несколько различных разработок на тему электронных таблиц. Но только в последние годы появились версии, показывающие реальную мощность электронных таблиц. Одной из таких разработок является Quattro Pro.

QUATTRO PRO 3.0: все, что нужно для продуктивной работы

Первая версия Quattro появилась в 1987 году и обладала практически теми же возможностями, что и электронные таблицы других фирм при цене почти в два раза ниже. Quattro Pro появилась в октябре 1989 года и завоевала 21 награду, вручаемую компьютерными журналами и ассоциациями.

Общие сведения о Quattro Pro

Как и большинство электронных таблиц своего класса, Quattro Pro поддерживает графические функции и связь с базами данных. Таблица может содержать до 8192 рядов и 256 колонок. Каждая ячейка может содержать текст, числовое значение, формулу или функцию.

Для доступа к командам меню могут использоваться либо последовательность клавиш, начиная с символа "/", либо клавиши управления курсором, либо мышь.

Так как Quattro Pro обладает большим числом стандартных и уникальных свойств, ниже они будут рассмотрены по категориям.

Возможности электронной таблицы

Одним из основных недостатков большинства электронных таблиц является невозможность объединения информации из разных файлов. Другими словами, если есть таблицы с данными о продажах по отдельным районам, то нет возможности объединить эти данные в суммарную таблицу. Quattro Pro решает эту проблему, позволяя одновременно загружать до 32 таблиц и устанавливать связи между их элементами. Для загрузки дополнительных таблиц используется команда меню File Open или File New, а для перемещения между таблицами — клавиша Shift-F6.

При загрузке более одной таблицы возникает проблема просмотра данных в таблицах. Quattro Pro позволяет просматривать таблицы различными способами. Например, если в памяти находятся три таблицы, их можно расположить таким образом, что будет видна половина первой таблицы и

по одной четверти двух других. Для полноэкранного просмотра таблицы можно использовать команду Window Zoom.

Для связи различных таблиц используется связующая формула, в которой указываются имя таблицы и адрес ячейки. Например, для связи с ячейкой F20 таблицы REGION1 используется следующая формула:

```
+[REGION1]F20
```

Также можно использовать специальный символ (*) для обозначения связи. Например, добавить значение, хранимое в ячейке F6, во все открытые таблицы можно при помощи формулы

```
@SUM([F6])
```

где SUM — функция суммирования; @ — символ связи; F6 — адрес ячейки.

В Quattro Pro также можно использовать связи произвольной формы. Например, можно создать следующую формулу:

```
+[REGION1](D3+G3)-
+[REGION2]E2
```

В эту формулу включена информация из двух ячеек таблицы REGION1 и одной ячейки таблицы REGION2.

При одновременной работе с более чем одной таблицей Quattro Pro предоставляет большой диапазон возможностей, от простых — функций изменения ширины колонки — до сложных — типа подсчета линейной регрессии и линейного программирования. Например, для подсчета линейной регрессии используются команды Tools, Advanced Math и Regression. Список функций вызывается клавишей Alt-F3.

Quattro Pro поддерживает выполнение формул и в обратную сторону — Solve For, при использовании которого сначала указывается желаемый результат, а затем подбираются необходимые для получения этого результата значения в формуле.

Также поддерживается возможность просмотра таблицы в режиме 132 колонок и 60 рядов. Этот режим отображает больше данных, чем стандартный, но предназначен он для видеоадаптеров TGA/VGA.

В Quattro Pro имеется возможность записи последовательностей команд (макросов) — так называемое средство Record. Для записи макроса необходимо выполнить последовательно команды Tools, Macro и Record. Это средство позволяет записывать все выполняемые действия до тех пор, пока запись не будет отменена. Командой Paste записанный макрос можно сохранить в таблице. Помимо стандартных команд, макросы могут содержать элементы программирования типа FOR, IF и BRANCH.

Если макрос неправильно работает, имеется возможность использования отладчика, который вызывается командами Tools, Macro и Debugger. Для ряда команд меню имеются управляющие клавиши. Для вызова отладчика используется команда Shift-F2.

Еще одной особенностью таблицы является архивирующее средство, называемое SQZ!. Оно позволяет экономить место на диске путем сжатия файлов-таблиц по специальному алгоритму. Это средство вызывается командами File, Utilities, SQZ!. Его эффективность крайне высока — например, таблица размером 140 Кбайт может быть сжата до 25 Кбайт.

После сжатия таблицы может возникнуть потребность в выводе ее на принтер. Quattro Pro позволяет оформлять выводимые на печать данные разными способами. Например, можно заключить таблицу в одинарную или двойную рамку с помощью команд Style и Line Drawing. Затем можно изменить шрифт в любой части таблицы (команды Style и Font). При помощи команды Screen Preview можно перед печатью посмотреть на экране, как таблица будет выглядеть на бумаге.

В Quattro Pro используются шрифты Bitstream (поставляется девять шрифтов) и поддерживаются шрифты для принтеров HP LaserJet и Apple LaserWriter. Кегль шрифтов может варьироваться от 6 до 72 пунктов.

При необходимости распечатки таблицы в развернутом на 90 градусов виде (sideways) используются команды Print, Layout, Orientation, Landscape. В этом случае принтер будет работать в графическом режиме, и распечатка будет занимать больше времени, чем обычно. Однако такой режим незаменим при печати таблиц, содержащих большое число колонок.

Отметим, что перечисленные выше свойства типа сжатия таблиц, изменения шрифтов и распечатки таблицы в повернутом виде доступны в других электронных таблицах только при использовании дополнительных утилит.

Графические возможности

Quattro Pro включает широкий набор типов графиков, в который входят текстовые диаграммы и три вида 3-мерных графиков. Quattro Pro содержит графический редактор Annotator с богатым выбором инструментов.

Можно изменять форму или размер графика с помощью мыши, добавлять текстовые окна, строки, заменять цвета, шрифты, размеры, вырезать и вставлять элементы из разных графиков, импортировать рисунки.

Одновременно в электронную таблицу может быть вставлено и распечатано до восьми графиков.

Наряду с графикой в Quattro Pro существуют и другие интересные возможности. Некоторые из них позволяют получить новые знания, исходя из уже имеющейся информации. Например, можно использовать функцию PMT для расчета максимальных ежемесячных выплат.

В Quattro Pro также есть поддержка 132-колоночного отображения и улучшенная сетевая организация.

Связь с Paradox

Quattro Pro может связываться напрямую с реляционной базой

данных Paradox, хотя эта функция еще далека от совершенства. Для этого необходимо предпринять следующие действия.

Запустить Paradox со специальным ключом в командной строке. Затем при выходе из Paradox запустить Quattro Pro. Если перед этим была специфицирована автозагрузка таблиц Paradox в Quattro Pro, они будут там восстановлены.

На этом возможности связывания исчерпываются. Другими словами, это не та динамическая связь, которая используется в Windows DDE. Нет возможности реализовать обратную связь файла электронной таблицы Quattro Pro и таблицы Paradox.

Операцию связывания можно осуществить, соединив 3 или 4 строки PAL-кода в Paradox с аналогичным макросом в Quattro Pro.

Сетевая поддержка

В Quattro Pro сетевая организация обеспечена тем же способом, что и в Paradox. В системном каталоге находятся файлы программ. Данный каталог является общим, и каждый пользователь может создать свой собственный каталог с индивидуальным доступом и средой.

Новым является и то, что файлы Bitstream-шрифтов в Quattro Pro могут использоваться коллективно, вместо того чтобы каждый раз копироваться пользователями.

Безопасность в сети

Файлы Quattro Pro могут быть защищены паролем. Они блокируются автоматически. Это удобнее, чем в системе ограничений Lotus. Но предлагаемая система менее гибка. В Lotus также возможно автоматическое блокирование файлов, в добавление к этому позволено вручную блокировать файлы и среду.

Заключение

Для удобства пользователей пакета Lotus 1-2-3 фирма Borland сделала очень многое. Quattro Pro не только восстанавливает рабочие таблицы, но и читает макросы Lotus. Есть даже специальный режим Lotus-compatible, который позволяет пользоваться меню, подобными Lotus.

Quattro Pro включает все передовые возможности электронных таблиц, библиотеки макросов, издательские функции, установку экранных цветов, поддержку импорта/экспорта различных форматов (DBase, Paradox, Lotus, Symphony и других).

В Quattro Pro включены встроенный архиватор SQZ, возможность выбора стиля меню (Lotus, Quattro Pro 1.0 или 2.0), возможность связи с Paradox, отладчик макросов, протокол команд и графический редактор Annotator.

Quattro Pro имеет также удобную сетевую организацию и графические возможности.

Техническая спецификация

Для эффективной работы Quattro Pro необходима операционная система MS/PC-DOS версии 2.0 или выше, 512 Кбайт оперативной памяти и память на жестком диске.

Quattro Pro поддерживает сети Novell Netware 2.0 или выше, 3Com Plus 1.0 или выше и 100%-совместимые с ними.

А. Федоров



Программные и аппаратные средства защиты информации для персональных компьютеров

Специализированные системы аппаратной и программной защиты можно разделить достаточно условно на три большие области:

- запрещение неавторизованного внешнего доступа к работающему компьютеру (со стороны червяков, вирусов);
- запрещение неавторизованного внутреннего доступа к отдельным файлам работающего компьютера, возникающего в результате случайного или преднамеренного подглядывания служащими;
- защита активных и архивных файлов, связанная с необслуживанием или отключением компьютеров из сети.

Последней мы не будем заниматься, поскольку это, в основном, разные варианты фильтров электрической сети и непрерываемых источников питания, сигнальные устройства, и для нашей темы рассмотрение этих устройств явно излишне.

Отложим на потом рассмотрение защиты от вирусов, червяков, чтобы уделить этому максимум места и внимания. Начнем с более традиционных вещей.

Максимальный уровень безопасности против всякого "подглядывания" обеспечивается запрещением неавторизованного доступа к компьютеру. Как сказал один из специалистов по компьютерной безопасности, любая такая система базируется на том, что "любому пользователю дается понять, что в случае нарушений его обязательно найдут и переломают ему ноги". Однако в системах, обеспечивающих доступ к компьютеру удаленному пользователю, проблема состоит в том, чтобы получить доказательства того, что пользователь именно тот, за кого он себя выдает (или кем себя называет).

Такие доказательства — это или вариации на тему парольной защиты, или различные биометрические системы. Для удаленных пользователей сейчас наиболее распространены системы "обратного вызова". Эти устройства при первом доступе к ним внешнего пользователя не дают контакта с вызываемым компьютером, а требуют назвать пароль или еще каким-нибудь образом идентифицировать себя. Если пароль назван правильно или идентификация пользователя успешна, ему возвращается номер телефона, ассоциированный с этим пользователем. Устройство обратного вызова запрашивает указанный номер телефона для подключения и выполнения нормальной процедуры login. Такие устройства выпускает, например, фирма B&B Electronics в Оттаве. Схемы "раздачи" паролей или обмен ключами легальных пользователей в развитых сетях могут быть очень изощренными и обеспечивать самые разные потребности пользователей в безопасном общении (телеконференции). Реализации таких схем приводят к появлению в сети специализированных

серверов паролей (проект CERBERUS). В то же время парольная защита характеризуется определенными недостатками, связанными с особенностями человеческой психики, поэтому фирмы пытаются найти другие способы идентификации пользователя, не связанные с указанием пароля. Эти аппаратно-программные системы проверяют самые разные биометрические характеристики (подпись, узор линий на пальце, узор сосудов на глазном дне, манеру работы на клавиатуре) для того, чтобы обеспечить эффективные гарантии идентификации локального или удаленного пользователя. Одной из популярных в правительственных учреждениях США является карта COMSPEC-II фирмы American Computer Security, которая обеспечивает интерфейс с самыми различными биометрическими устройствами типа электронных подписей, идентификаторов кожного узора, а также контроль доступа к каналам больших компьютеров, шифрование архивных и резервных копий данных, защиту от копирования локальных файлов. Эта плата стоит довольно дорого, однако программное обеспечение в комплекте с этой платой — на порядок дороже. К сожалению, более подробная информация об особенностях этого интересного оборудования и программного обеспечения автору неизвестна.

Большинство таких систем обладают тем недостатком, что не отслеживают непрерывно во время работы неизменность биометрических характеристик пользователя, поэтому допускают в принципе временную "подмену" пользователя без какой-либо реакции со стороны системы защиты. Конечно, определенные меры, затрудняющие "подмену", принимаются: пользователь может "запереть" компьютер, отлучаясь ненадолго, для "отпирания" нужно знать специальный пароль; после определенного перерыва в непрерывной работе пользователь автоматически отключается от системы и для подключения ему нужно снова предъявлять свои полномочия. Далее мы подробно поговорим о двух наиболее интересных системах защиты. Первая — чисто программная, но позволяет в одной из своих модификаций непрерывно отслеживать, кто работает за компьютером; вторая — аппаратно-программная и отличается продуманностью.

1. Фирма Electronic Signature Lock из Орегона поставляет пакет программ ESL, который анализирует манеру пользователя работать на клавиатуре во время набора своего имени и пароля (парольная защита не исключается полностью!). Конечно, эта манера изменяется со временем или зависит от состояния здоровья и т.д. Утверждается, что используемые статистические алгоритмы столь изощренны, что позволяют учитывать эти неизбежные вариации. Фирма утверждает, что вероятность неверной идентификации пользователя, если он знает правильный пароль, менее 10^{-6}. Существует более совершенный вариант пакета CESL, который непрерывно отслеживает манеру пользователя работать на клавиатуре и может определить, когда за клавиатурой оказался не тот человек, что предъявлял полномочия на доступ. К сожалению, информация об этом пакете недостаточно подробна: нет даже намеков на то, во что выливается это слежение по части машинных ресурсов, насколько это усложняет работу пользователя, насколько пакет совместим с другим программным обеспечением.

2. Разработка фирмы Datamedia. Серия ее компьютеров Netmate оборудована специальным устройством Securecard reader — считывателем карт безопасности. Применение таких считывателей довольно популярно на Западе, например, в PC-Watchmane используется считыватель обычных кредитных или ID-карт для авторизации пользователя в дополнение к паролю. Карты безопасности по исполнению — вариант кредитных карт; на них на магнитном носителе с помощью специальной аппаратуры, которая имеется только в распоряжении администратора, делается запись о пользователе: его имя, пароль и описываются все полномочия, которые он получает при входе в систему. В частности, на карте записано, сколько раз пользователь может пытаться указать пароль при входе. Таким образом, случайная потеря карты безопасности или ее кража не позволяет злоумышленнику получить доступ к компьютеру: если имя пользователя еще можно узнать не привлекая внимания, то пароль ему неизвестен. Только сознательная передача карты безопасности кому-то одновременно с разглашением пароля может открыть доступ к компьютеру постороннему лицу.

Администратор системы создает карту безопасности для легальных пользователей. На этой карте, помимо уже перечисленной информации, описывается профиль пользователя. В него включаются, например: возможность доступа к программе SETUP, то есть фиксируются такие характеристики компьютера, как экран, количество и типы дисков; также определяется, какие из локальных устройств (гибкие диски, жесткие диски, последовательные и параллельные порты) доступны этому пользователю, с каких локальных или сетевых устройств он может загружаться. Предусмотрена и трансляция паролей: тот пароль, который назначается пользователю, как правило, легко запоминающийся, но вовсе не тот, с которым работает система.

В программное обеспечение встроены специальные возможности, которые повышают уровень "тревожности" в случае возникновения ненормальных ситуаций. При попытке просто выдернуть карту безопасности из считывателя — доступ к компьютеру намертво блокируется, пока в считыватель не будет вставлена та же карта безопасности. И если такая ошибка для легального пользователя — несущественна, для злоумышленника такой оборот дела совершенно неприемлем. Так всячески стимулируется желание администрации до конца разобраться в необычной ситуации, а не прятать сор в темном углу. При неправильном указании пароля (если превышено количество попыток, разрешенное для данного пользователя) — машина также блокируется, и только администратор сможет "оживить" ее, то есть опять

стимулируется необходимость довести до сведения администрации все случаи нарушения режима секретности.

С точки зрения защиты от вирусов перечисленные системы, конечно, важны, поскольку они, кроме идентификации пользователя, определенным образом организуют его работу на компьютере, запрещая отдельные опасные действия типа запуска программ с дискеты, загрузки с дискет. Ограничения на использование определенных ресурсов системы типа сетевых карт, последовательных портов, так же полезны с точки зрения защиты от вирусов, поскольку ограничивают возможность или даже отсекают некоторые пути распространения или "получения" заразы. Наконец, повышенный уровень тревоги, характерный для некоторых систем, очень полезен и с антивирусной точки зрения: любые неполадки и странности в работе компьютеров немедленно должны становиться достоянием администрации и так же немедленно доводиться до сведения специалистов, — это резко уменьшает размеры ущерба от проникновения вирусов. И все-таки эффективность этих средств с точки зрения защиты от вирусов ограниченна.

Очень простым, но весьма эффективным средством обеспечения безопасности данных и защиты от вирусов является специальная конструкция компьютера, благодаря которой воровство данных или заражение файлов или файловых систем вирусами становится невозможным. Так например, фирма Earth Computer из Калифорнии делает серию компьютеров EarthStation, приспособленных для работы в локальной сети, но не имеющей шинных разъемов, то есть принципиально нерасширяемых (изготовителей компьютеров такого типа довольно много во всем мире). Главная (и единственная!) плата этого компьютера расположена в клавиатуре. Управление клавиатурой, видеоадаптер и сетевой адаптер объединены на этой плате. Даже монитор предлагается покупать отдельно. В компьютере находится специальный вариант BIOS, позволяющий загружать DOS по локальной сети. Нельзя отказать разработчикам таких компьютеров в остроумии: если некуда положить ворованное, то лучше и не красть; относительно заражения вирусами проблема решается так же, как с воровством, — на компьютере нечего заражать. С червяками, конечно, сложнее: они могут проникать на этот "уродец" и безусловно могут отбирать ресурсы на собственное размножение.

Большинство фирм, однако, защищают свое оборудование с помощью одного из вариантов парольной защиты более или менее интегрированной с оборудованием. К их числу относятся, например, фирмы Compaq и Hewlett-Packard. Правда, для них этот вид деятельности не является основным, и те варианты защиты, которые фирмы предоставляют клиентам, не слишком совершенны (особенно это относится к компьютерам Vectra фирмы Hewlett-Packard). Разработка Compaq упоминается здесь лишь из уважения к фирме и потому, что похожие элементы защиты часто встречаются в компьютерах различных фирм. Compaq разработала для своих моделей DeskPro специальную версию BIOS, содержащую элементы защиты от несанкционированного использования компьютера. Так, при загрузке компьютера при включении питания еще во время процедуры POST требуется указать правильный пароль, чтобы машина продолжала работу (такую возможность защиты компьютера предоставляют многие фирмы, более того, часто возможность реализована в BIOS, но не описана в документации, некоторые вирусы могут записывать в поле пароля случайную информацию, и однажды пользователь обнаруживает, что его машина неплохо защищена от него). Сам пароль хранится в области CMOS и при большом желании, безусловно, может быть стерт (фирма Hewlett-Packard предусматривает хранение в специальной области CMOS предыдущей конфигурации компьютера, и по нажатию кнопки эта конфигурация может быть восстановлена в обычной области CMOS), но для затруднения доступа к "внутренностям" компьютера фирма снабжает корпус качественным замком (физический уровень защиты). Кроме этой возможности в BIOS реализованы программы, поддерживающие следующие области разделения доступа: возможность быстрого запирания компьютера, защита серверного состояния компьютера, защита жесткого диска, гибкого диска, последовательного и параллельного портов (и эти возможности защиты реализованы только в BIOS фирмы Compaq). Запуск защитных программ из BIOS регулируется переключателями на плате компьютера (аппаратный уровень защиты).

Упомянутые выше аппаратные доработки плат, усовершенствования BIOS, конечно, нужно всемерно приветствовать, поскольку они улучшают условия работы на персональном компьютере с точки зрения безопасности, приучают пользователя к "порядку" и одновременно ограничивают возможность случайного заражения, но проблему комплексной защиты от вирусов, увы не решают.

Защита от вирусов тесно соприкасается с задачей защиты информации от несанкционированного доступа, хотя это и не одно и то же. Обе эти задачи затрагивают тему удобства работы на компьютере. Как правило, легче осложнить себе жизнь, чем хорошо защититься, но, видимо, любая защита так или иначе связана с определенными неудобствами в пользовании компьютером.

Важны и чисто человеческие аспекты защиты. Например, в системе разграничения доступа NetWare не доступные пользователю части файловой системы ему и не показывают, а в системе разграничения доступа WatchDog — показывают, но при попытке получить доступ к этим областям требуют пароль. Очевидно, что в этом смысле NetWare — ближе человеческой психике и совершеннее — отсутствует элемент провокации пользователя, имеющийся в WatchDog.

Система разграничения доступа NetWare — чисто программная, очень продуманная как с точки зрения

администратора: легко и удобно вводить нового пользователя в систему, создавая для него систему полномочий и областей доступа; так и с точки зрения пользователя: разграничение доступа достаточно "прозрачно" во время работы и, к тому же, надежно разделяет пользователей. О степени продуманности системы разграничения доступа говорит, например, следующая возможность: закрыть для определенной категории пользователей часть файловой системы в некоторое время суток, дни недели и т.д. Так естественно решается проблема игр: хочешь играть — задерживайся после работы. Эта система разграничения доступа довольно долго "держалась" под напором хакеров, что, видимо, связано с относительно малой популярностью NetWare (по сравнению с DOS) и относительной неразвитостью локальных сетей под управлением NetWare (сети UNIX компьютеров соединяют десятки, сотни и тысячи серверов; для NetWare количество серверов в локальной сети редко превышает несколько штук).

По сообщению PC WEEK Vol. 7, No. 44, от 5 ноября 1990 года, появился сетевой вирус в среде NetWare, который обходит защиту от записи на сервере, позволяя писать на сервер с узлов сети, не обладающих такими правами. По словам PC WEEK, фирма Novell пока не распространяет по этому поводу никаких подробностей. Более того, очевидно, проблема обеспечения безопасности в сетях далека от разрешения. Летом этого года в LAN Times появилось много публикаций о пакетах, повышающих безопасность, совершенствующих разделение доступа, отслеживающих состояние сетей Novell. У некоторых пакетов отчетливый антивирусный уклон, поскольку они могут, например, запретить запуск или копирование программ с локальных дисков.

Другой системой разграничения доступа, с которой автору пришлось работать, является WATCHDOG, изготавливаемая фирмой Fisher International Systems из Флориды.

Система претендует на некую универсальность, во всяком случае фирма рассматривает в документации возможности применения WatchDog в средах UNIX, OS/2, Windows, DOS, в дополнении к NetWare. Существуют варианты системы, поддерживаемые аппаратно.

Первоначально система рассматривалась фирмой только как система разграничения доступа, но с "расцветом" компьютерных вирусов система предлагается и как средство антивирусной защиты. Впрочем, ничего специального антивирусного эта система разграничения доступа не содержит: присутствует более жесткий дополнительный контроль попыток писать в отдельные области, который все же определенным образом сдерживает и обнаруживает размножение вирусов.

Система очень продуманна с точки зрения повышения безопасности: предусмотрены возможность "насильственной" смены пароля каждые несколько дней; проверка нового пароля на "новизну" (нельзя использовать пароль, уже бывший в употреблении на конкретном компьютере); возможность быстрого блокирования и разблокирования клавиатуры по предъявлению особого пароля, когда пользователь в процессе работы ненадолго отлучается, и стало быть, есть вероятность работы, пусть недолгой, неавторизованного лица, если какое-то время пользователь не работает на клавиатуре, — он насильственно "выгоняется" из WatchDog´a; возможность разделения файловой системы на логические области, вход в которые осуществляется по предъявлению специальных полномочий; возможность использовать различные программные средства (в основном развитые) по предъявлению специальных полномочий. Система продуманна в отношении трудности "вскрытия" защиты. Основным ее элементом является драйвер, использующий для своей работы информацию, сосредоточенную в нескольких десятках файлов, располагаемых WatchDog на диске.

Продуманность секретности, однако, не сочетается с удобством работы для пользователя, хотя фирма и предлагает использовать ее генератор меню. Кстати, многие системы разграничения доступа пытаются как-то компенсировать неудобства, причиняемые пользователю системой разграничения, с помощью разнообразных меню. Этим пользуются и WatchDog, и PC-Watchman, и NetWare, и Disk Oganaser, и многие другие изготовители программ для разграничения доступа.

Хотелось бы отметить, что использование любого генератора меню или работа в среде такого генератора, пусть даже не рассчитанного на серьезное разграничение доступа, повышает "вирусоустойчивость". Дело в том, что пользователю "ненавязчиво навязывается" определенный стиль работы, неявно ограничивается круг программ, с которым происходит работа, создаются помехи или неудобства для запуска программ с гибких дисков, что резко снижает вероятность непреднамеренного заражения (см. например, описание Lazy Susan, материалы фирмы Getc Software).

Идеологически WatchDog работает так: изменяет форматы системных таблиц на жестком диске так, что обычная система(-ы) не в состоянии опознать и работать с таким жестким диском. Сама WatchDog работает с измененными таблицами, используя драйвер, который корректирует таблицы, но требует прежде предъявления имени и пароля для доступа к ресурсам компьютера. Как уже упоминалось, существуют различные уровни защиты. Создатели WatchDog понимают, что изменения в системных таблицах не являются настоящим препятствием для хакера. Автор имел опыт общения с WatchDog такого рода: однажды в результате непродуманных действий был заперт жесткий диск. После нескольких дней работы, не зная о WatchDog ничего, удалось открыть файловую систему на жестком диске для чтения. Теперь уже можно проделать такую же работу за час-полтора.

Поэтому в системе предусмотрена и более серьезная мера защиты информации: некоторые области могут быть зашифрованы с применением DES-алгоритма.

Аппаратные средства, которые могут быть использованы вместе с WatchDog, состоят из одной платы

WatchDog ArmorTM. Все, что обеспечивает это оборудование, сводится к следующему:

- запрещение работы с дискетами до тех пор, пока не произойдет корректная загрузка с жесткого диска с установленным там WatchDog;
- реализация DES-алгоритма аппаратно, с помощью DES микросхемы, что увеличивает скорость работы с применением этого уровня защиты;
- наличие независимых часов, обеспечивающих точную временную привязку деятельности WatchDog и недоступных для неавторизованного изменения.

Следует отметить, что при всей привлекательности и трудности преодоления защита с помощью шифрования обладает некоторыми негативными чертами, которые в нашей стране особенно существенны. Применение этого уровня защиты означает для персонала (видимо, это персонал, ответственный за обеспечение секретности) новую жизнь, полную жестко регламентированных предопределенных действий. Необходимо обеспечить ежедневную службу backup, поскольку ошибки в чтении отдельных секторов или кластеров защищенной шифрованием области могут приводить к потере информации из всей области. Отсюда повышенные требования к качеству дисков, на которых расположена шифрованная область; абсолютные требования к качеству и регулярности службы backup; повышенный риск, по меньшей мере, дополнительные потери времени на восстановление информации в случае сбоев. Эти факторы существенны при работе с любым программным обеспечением или аппаратурой, обеспечивающей криптографический уровень защиты данных, и не связаны с особенностями пакета WatchDog. Рекомендовать применение такой защиты можно только при наличии качественного, надежно работающего оборудования, и желательно организовать надежную, автоматическую службу backup и наличии действительной потребности в столь сложной защите данных.

В то же время, использование криптографической техники многие эксперты по компьютерной безопасности считают основным, если не последним бастионом защиты от вирусной инфекции, во всяком случае, применение шифрования может сильно затруднить распространение вирусов. Это относится к любым вирусам, присутствие которых криптографические программы могут обнаружить по изменениям контрольных сумм файлов и других характеристик. С другой стороны, это защита, срабатывающая, как правило, после того как заражение уже состоялось, и создающая определенные неудобства пользователям, а также персоналу, ответственному за обеспечение безопасности.

Частным случаем систем разграничения доступа являются системы защиты программного обеспечения от незаконного копирования. Автор затрагивает эту тему в обзоре, поскольку в этой области имеются разные аппаратные разработки, не просто интересные, но в случае развития могущие серьезно изменить подход к проблеме разграничения доступа и защиты от вирусов и червяков.

Начну издалека. Программные системы защиты и "плохие" аппаратные средства защиты от несанкционированного копирования так или иначе сводятся к проверке каких-то специфических условий или параметров наличной аппаратуры или носителей информации. "Вскрытие" таких систем защиты — вопрос времени и опыта программиста. Рано или поздно удастся найти критические команды, с изменением которых уничтожается вся система защиты. Все ухищрения, к которым прибегают разработчики (например, шифруют критические участки программы, раскрывая их непосредственно перед исполнением), — паллиатив, который лишь затягивает дело. Чаще всего даже нет необходимости в подробном и глубоком анализе: например, если в какой-то момент в памяти оказывается корректный исполняемый модуль — проще всего записать его образ на носитель с помощью подходящей резидентной программы, даже не задумываясь о сложностях защиты.

Самым "естественным" способом использования аппаратуры для защиты программ от несанкционированного копирования является, наверное, широкое использование особенностей оборудования при написании программы. Эта "естественность", однако, может входить в противоречия с требованиями универсальности и простоты тиражирования программного обеспечения.

"Хорошая" защита строится по типу "черного ящика". Оборудование определенным, но уникальным образом программируется; соответственно изменяется и код программы в определенных местах; в некоторых точках программы происходит обращение к оборудованию с известными программе параметрами; оборудование возвращает какие-то данные, которые используются, например, для построения кода какого-то участка программы. Такая схема трудна для "вскрытия", поскольку невозможно установить по данным, передаваемым оборудованию, и по данным, получаемым от него в ответ, что за программа работает внутри. Сравнение разных копий дает мало информации для понимания происходящего (как правило, разные копии "зашумлены" случайными изменениями, чтобы еще более затруднить поиск реальных отличий). Наконец, локальный характер обращений к оборудованию позволяет ни в один момент не строить всего исполняемого кода.

Обычный вопрос при построении таких систем "хорошей" защиты: как обеспечить независимое функционирование оборудования, легкость монтирования и размонтирования. Первоначально такая аппаратура размещалась внутри компьютера. Так, ранние версии Novell защищались с помощью keycard, вставлявшейся в разъем шины сервера. Такое решение нельзя признать очень удачным: для монтирования-размонтирования нужно открывать компьютер; понапрасну теряется дефицитный разъем.

Современная аппаратура защиты располагается, подключаясь к наружным разъемам параллельного или

последовательного порта. Рекламируют подобную продукцию многие, но неясно, что это: собственные оригинальные разработки, содержащие нетривиальные идеи, или вариации разработок одной фирмы. Во всяком случае, представители фирмы FAST Electronic Gmbh. утверждали, что кристаллы, используемые в их изделии, очень сложны и изготавливаются двумя-тремя фирмами в мире.

Использование изделий, аналогичных продуктам фирмы FAST, позволяет любое программное обеспечение — от операционных систем до пакетов прикладных программ — быстро и эффективно сделать уникальным. То есть сильно затруднить универсальную порчу кода аналогичного программного обеспечения и, тем самым, блокировать репликацию системных вирусов, распространение червяков или, по крайней мере, вызвать остановку или зависание операционной системы, что является "хорошим" сигналом для пользователя о неблагополучной ситуации. Универсальная возможность изменения работы остается на уровне функций, а это все же поддается контролю. К сожалению, таким образом невозможно блокировать "деятельность" вирусов в прикладных пакетах, но использование средств более развитых, чем нынешние, дает хорошую основу для контроля неизменности кода прикладного пакета, без чего немыслима защита от вирусов в прикладных пакетах.

Переходя к специализированным аппаратным средствами защиты от вирусов, червяков и прочего вредоносного программного обеспечения, хотелось бы отметить следующее: все эти средства рассчитаны на борьбу с системными вирусами и, к сожалению, информация скудна, а та, которая есть, — вся о защите от вирусов в DOS. Между тем, возможно написание вирусов и на языке прикладных программ; известны вирусы в среде Lotus 1-2-3, на языке BAT-файлов в DOS; известны вирусы на языке Shell в Unix (Xenix). То, что это исследовательские вирусы, — существа дела не меняет, настолько они просты и естественны. То есть могут легко появиться и в результате сознательных действий, и в результате неосознанной ошибки. Аппаратуры для борьбы с такими вирусами, конечно, никто никогда не создавал и, если аппаратные средства как-то защищают от такого рода вирусов, то это побочный результат работы соответствующей защиты.

Речь идет о том, что мы называем "проблемой копирования". Любая программная система должна включать какие-то средства для копирования текстов, файлов, данных ... Естественно стремление к тому, чтобы указанное средство было удобным, это немедленно порождает вирусные ситуации. Задача саморепликации любого текста сильно упрощается, если имеется доступ к этому самому тексту. Практически для всех современных операционных систем стандартом является передача в самом первом параметре имени исполняемого файла, то есть доступ к тексту! Если для DOS имеются осложнения технического характера, связанные с соглашениями об именах исполняемых файлов, которые не позволяют написать коротенький универсальный самореплицирующийся код, то для Unix — эта задача тривиальна. Например, код

```
cp $0 $1
```

прекрасно вставляется в любой командный файл и делает этот файл при запуске с параметром размножающимся, то есть вирусным. Какое оборудование может препятствовать размножению таких командных файлов?! С точки зрения системы все совершенно корректно, единственно, что может вызвать какие-то возражения у системы, — место в файловой системе, куда указано записать копию командного файла.

Но вернемся к аппаратным средствам защиты от системных вирусов в DOS.

Аппаратные средства защиты от вирусов можно разделить на системно-зависимые и системно-независимые. Неявно мы, конечно, ведем речь об IBM-совместимых персональных компьютерах, для которых защита от вирусов — наиболее острый вопрос; с другой стороны, массовость этого типа компьютеров делает оправданными усилия по разработке специализированных аппаратных средств защиты от вирусов. Более того, для других типов компьютеров (Macintosh, рабочие станции, серверы, mainframe под операционными системами типа UNIX) разработка системно-независимых аппаратных средств защиты, очевидно, затруднена — уж очень подвижна и "развесиста" файловая система. Операционные же системы на этих компьютерах работают, как правило, в защищенном режиме, в который переключают процессор на самых ранних этапах загрузки, а это дает какую-то гарантию неизменности операционной обстановки, да и сами операционные системы более жестко проверяют состояние своих внутренних областей: векторов прерываний, буферов, системных таблиц. Для таких компьютеров и операционных систем многие проблемы возникают во многом из-за неадекватного состояния файловой системы. Наличие файлов и каталогов, открытых для записи кому угодно, программ, которые может использовать кто угодно, отсутствие авторизации или короткие пароли пользователей. Эти проблемы решаются различными программными инструментами администратора системы: пакеты UXA, Alert/Inform, Quest.

Известную аппаратную защиту от вирусов в DOS можно разделить на два типа:
- перлюстрация прерывания 21;
- перлюстрация команд к физическому устройству.

Первая, очевидно, только системно-зависима; вторая — может быть и системно-зависимой, и системно-независимой.

Понятно, что защита первого типа — более гибкая и удобная с точки зрения пользователя, но ее легче обмануть и вирусу; защита второго типа — менее гибкая и более надежная и универсальная.

Защита второго типа, по нашему мнению, наиболее перспективна для защиты от вирусов и с точки зрения надежности, и системно-независимые варианты защиты — с точки зрения универсальности и по отношению к системному обеспечению, и к прикладному. Накладные расходы на работу специфической аппаратуры тоже приемлемые. Есть претензии по части гибкости защиты, но при правильном отборе программного обеспечения, при разумных затратах на настройку файловой системы и программного обеспечения результаты для персональных компьютеров прогнозируются хорошие. Как уже говорилось, сложно представить себе использование такой схемы защиты в развитой конфигурации mini- или mainframe-компьютера при очень подвижной, разветвленной файловой системе, когда динамически монтируются/размонтируются многочисленные файловые системы.

Отметим очевидные достоинства и недостатки использования аппаратурной защиты первого типа для борьбы с вирусами.

1. Аппаратура, пока она защищает, является надежным барьером на пути инфекции (надежнее программных средств).

2. После того как аппаратуру научились "обманывать", перенастроить ее значительно дороже и сложнее, чем модифицировать программное обеспечение.

Что делают платы перлюстрации? Защита этого типа действует примерно так же, как действуют программные мониторы, то есть просматривают 21 прерывание и пытаются выявить незаконные действия, только перехват этого прерывания и дополнительная проверка условий запроса сервиса DOS реализованы аппаратно. Конечно, такая реализация позволяет избавиться от кое-каких проблем (например, некоторые программные мониторы и вирусы используют "поплавковый" механизм для более надежной привязки к 21 прерыванию, состоящий в том, что перехватываются еще некоторые прерывания, например, прерывания от системного таймера; когда такие программы работают на компьютере одновременно, могут возникать тупиковые ситуации, зависания и т.п.). Неприятность, однако, состоит в том, что сейчас вирусы часто используют прерывание 2Е или модифицируют код обработчика 21 прерывания DOS, и неясно, насколько адекватно аппаратура обрабатывает такие разные (и все более и более разные) ситуации. И уж, конечно, эта защита бессильна против макровирусов, то есть вирусов, живущих в среде прикладных пакетов: с точки зрения DOS все запросы функций 21 прерывания одинаково корректны.

Одной из лучших, по впечатлению очевидцев (В.Бончева), является плата Tender Byte Франца Вельдмана (Frans Veldman) из Нидерландов, которая реализует просмотр того же 21-го прерывания и поэтому более уязвима для программных усовершенствований вирусов.

И за рубежом, и у нас есть средства аппаратной защиты второго типа. Только по рекламе известны: Disk DefenderTM (Director Technologies, Inc.); Guard CardTM (NorthBank Corporation), программно-аппаратный комплекс Intellectual Write Protector (BIS Corporation).

Остановимся на Disk Defender, который реализует грубую, но действенную аппаратурную защиту. Плата Disk Defender представляет собой дополнительный контроллер, который располагается между стандартным MFM-контроллером и жестким диском. Эта плата защищает содержимое всего жесткого диска или его части, блокируя выдачу команд записи на жесткий диск. Управление защитой осуществляется с внешнего пульта управления, который подключается к плате и может размещаться довольно далеко (до 1,5 метров). Переключателями на панели пульта устанавливается область (в цилиндрах), защищаемая на диске, и вариант защиты: от доступа, от чтения, от записи. При отключении пульта действует последняя установка.

Поскольку плата осуществляет перлюстрацию физических команд, выдаваемых даже не на контроллер, а на физическое устройство, защита работает независимо от программного обеспечения и операционной системы, независимо от усилий, предпринимаемых вирусами, троянскими конями, червяками со стороны компьютера. Программа на компьютере может использовать системные вызовы, может использовать BIOS или даже непосредственно обращаться к контроллеру диска, работая с портами, — защита с легкостью блокирует все эти действия. Эта защита столь же надежна, как и применение только считываемых носителей, например, CD-ROM, но более удобна и совершенна. Защита, конечно, универсальна в том смысле, что блокирует распространение и системных, и макро- и любых других вирусов, но только в защищенных областях диска. Правда, заодно в этих областях становятся невозможны и правка текстовых файлов редактором, и трансляция программ, и многое-многое другое. Еще один недостаток этой защиты состоит в том, что защищаемая область — физическое место на диске, какие каталоги, файлы, какая часть файловой структуры окажется в этой физической области — вопрос кропотливой настройки файловой системы.

С другой стороны, защита в определенных ситуациях оказывается обременительной настолько, что, очевидно, ее необходимо отключать совершенно для каких-то областей диска или всего диска в целом (например, инсталляция новых продуктов, обновление и т.п.). И в этот момент все становится уязвимым, как на обычном компьютере. Конечно, во время таких операций следует проявлять повышенное внимание, аккуратность, но в современных условиях поражение вероятно, особенно если персонал не имеет навыка работы в вирусной среде (это — побочный результат хорошей работы аппаратной защиты). Правда, после включения защиты распространение вируса на защищенной части будет невозможно, что, вероятно, минимизирует возможный ущерб.

Недостаточная гибкость побуждает разработчиков аппаратной защиты второго типа создавать более совершенные варианты защиты. Например, отечественная разработка IWP (А.В. Водяник и др.) имеет развитые программные средства; другие проектируемые системы также предполагается сделать значительно более гибкими и программно управляемыми... Но немедленно теряется универсальность защиты по отношению к операционной среде: раз мы знаем что-то кроме физических адресов на диске — мы немедленно оказываемся связанными со стандартами той файловой системы, которую собираемся защищать. Наиболее сильная сторона аппаратной защиты второго типа — программное управление аппаратурой защиты, без чего немыслима гибкая, адаптивная защита носителя — парадоксальным образом оказывается самой слабой ее стороной. Как только мы открыли дорогу программному управлению аппаратурой, у нас исчезли гарантии того, что аппаратура будет функционировать только так, как нам того хочется. Пусть это маловероятно, но все-таки возможно программно отменить все аппаратные ограничения на обмен или перенастроить эти ограничения, поскольку это возможно на программном уровне. В конце концов, до появления вирусов любая операционная система работала с магнитными носителями без всяких ограничений. С появлением вирусов выяснилось, что возможны такие искажения программ операционной системы, которые внешне почти незаметны, но приводят к радикальному изменению системы с исполняемыми файлами и файловыми системами. Что мешает так же незаметно изменить функционирование программно-управляемой аппаратной защиты? Конечно, это специализированное оборудование, специальные программы для управления этим оборудованием, однако элемент физической непреодолимости защиты исчез.

Несмотря на все эти принципиальные "недостатки" в гибкой аппаратной защите, несколько слов о продукте IWP.

Имеется программный вариант защиты диска, состоящий из двух программ: IWP.EXE и MAP.EXE. Последняя программа предназначена для создания карты защищенных файлов, которая записывается в файл _IWP_.MAP в корневом каталоге (в файле фигурирует и запись в каталоге и цепочка FAT). Программа IWP.EXE — резидентная и следит за запросами на запись, проходящими через драйвер диска. Все запросы на запись проверяются по карте _IWP_.MAP на корректность и подавляются, если обнаружена попытка записи в закрытый файл. Заметим, что области на диске здесь уже логическое понятие, это правильно и хорошо, но в то же время любое изменение в файле _IWP_.MAP может привести к непредсказуемым последствиям. Перлюстрируются обращения в порты контроллера диска, и обращения не из BIOS считаются опасными. Аппаратная часть называется PWC (Port Watch Card). Имеются два варианта платы. Наиболее совершенный вариант позволяет использовать систему паролей для доступа к жесткому и гибким дискам, CMOS; в обоих случаях обращение к портам контролируется аппаратно. Пароли для более сложной платы могут быть перестроены программно (PWD.EXE).

Последнее, о чем хотелось бы упомянуть в контексте борьбы с вирусами, — это слежение за работой вычислительной системы. Слежение может быть разного уровня реализации (программное, аппаратное, программно-аппаратное), разного уровня подробности (обращения к физическим портам, адресам памяти, ресурсам; отслеживание логических ресурсов, использования программ, каналов и т.п.). Отслеживание, сбор статистики и обработка этой статистики не решают задачу активной обороны, но являются очень хорошим диагностическим средством. Обычно средства слежения фиксируют такое количество взаимосвязанных событий, что невозможно скрыть следы и проникновения в систему, и отключения системы слежения. В лучшем случае удается так исказить данные си-

стемы слежения, что они, на первый взгляд, выглядят корректными. Но при внимательном, подробном анализе всегда обнаруживаются маленькие шероховатости, неувязки, несогласованность данных. Поэтому сбор статистики оказывается основой, вокруг которой так или иначе концентрируются самые разные способы борьбы или обнаружения вирусов, пиратских проникновений в систему хакеров. Отношение к статистическим данным должно быть простым: крупные, кричащие несоответствия скорее всего говорят об ошибках персонала и особенной опасности не представляют; напротив, мелкие расхождения должны становиться предметом самого придирчивого расследования, поскольку они и являются, скорее всего, свидетельством сознательного нелегального проникновения в систему. Для операционной системы DOS сбор какой бы то ни было статистики не характерен (встроенных средств для этого в системе нет), для UNIX, VM, MVS и других многопользовательских систем, — это является стандартом. Некоторые генераторы меню в DOS (Lazy SUSAN, например) позволяют собирать статистику. Эта статистика не является столь исчерпывающей и многоплановой, как в многопользовательских системах, она и не может быть такой — сама "операционка" много проще, сложных событий и ситуаций просто не бывает. Но уже простейшая статистика, регистрирующая время начала работы, время запуска и время работы различных программ и процедур, дает хорошую основу для анализа работы системы. Поэтому различные системы отслеживания, сбора статистики, особенно специально спроектированные с целью аккуратной фиксации специфически вирусных (или подозрительных, квазивирусных) событий, при всех накладных расходах, которые они порождают, являются важным и полезным средством в борьбе за безопасность компьютерных систем. Особенно важными эти системы оказываются в локальных и глобальных сетях. Впрочем, в этой среде развитие систем сбора статистики, отслеживания событий стимулировалось задолго до начала вирусной эры и совсем другими потребностями, но, хоть и борьба с вирусами — побочный результат, — сами системы в борьбе против вирусов необычайно полезны.

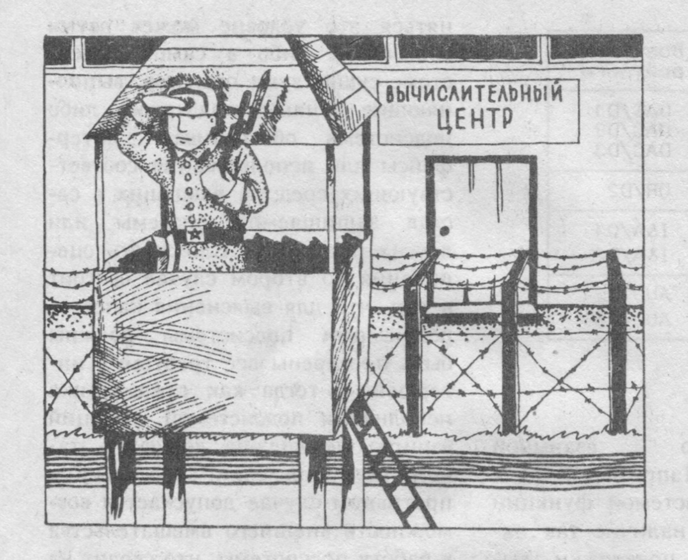
Любопытные данные можно найти в Computerworld Focus on Integration (June 5, 1989, Crime Time, Ann Dooley, pp. 30-32.). Там приведена таблица методов, которыми пользуются различные организации в США (то есть люди, действительно имеющие реальный выбор средств и могущие его делать в зависимости и от стоимости, и от стоящих задач) для защиты рабочих станций на базе PC или отдельных персональных компьютеров этих средств, составленная по результатам телефонного опроса 214 фирм. Речь идет, конечно, не о защите от вирусов, а о безопасности вообще, и все-таки... Вот эта таблица:

Парольная защита	99%
Слежение за работой	73%
Физическая изоляция компьютеров	63%
Остальные	24%
Аппаратная защита	13%
Модемы с обратным вызовом	6%
Различные методы опознания личности (сетчатка глаза, голос и т.д.)	4%

Как видно из этой таблицы, единственного, самого надежного метода нет; как правило, используется та или иная комбинация методов защиты, которая и дает, с точки зрения администрации, приемлемые результаты. Более того, относительная непопулярность аппаратной защиты говорит о плохом соотношении стоимость/надежность для аппаратных средств. И, напротив, такие относительно дешевые, простые средства, как минимальная парольная защита, физическая изоляция компьютеров, сбор статистики работы, оказываются достаточными с точки зрения администраторов систем. Если по отношению к двум последним мероприятиям возражений не возникает, — по отношению к парольной защите нужны определенный скептицизм и настороженность: как мы уже видели, сама по себе парольная защита проблему безопасности не решает, но лишь создает иллюзию защиты. Насколько тонким вопросом является применение парольной защиты, может показать следующая история. В одной из американских клиник был установлен компьютер. В связи с тем, что информация о состоянии здоровья там конфиденциальная, была установлена сложная парольная система разграничения доступа. Видимо, интерфейс с человеком был недостаточно продуман, и врачи решили проблему общения с компьютером следующим образом: к каждому терминалу был прикреплен листок с инструкцией, в которой описывалась процедура вхождения в систему и был приведен полный список паролей...

Между тем парольная защита становится все популярнее. Фирма Helett-Packard недавно анонсировала продолжение своей серии Vectra как компьютеров для работы "без проблем". Важным элементом этой серии является двухуровневая парольная защита, встроенная в BIOS, а это, как мы видели, создает хороший базис, но отнюдь не решает всех проблем парольной защиты.

И. Карасик



Эта статья продолжает тему компьютерной безопасности, начатую КомпьютерПресс в №№10-12, 1991 г. и №2, 1992 г. В ней рассказывается о том, как обстоят дела с безопасностью в одной из самых "компьютерных" стран мира — Соединенных Штатах Америки, и как сами американцы оценивают надежность и безопасность своих компьютеров и програм.

Американская классификация и принципы оценивания безопасности компьютерных систем

Подсистемы оцениваются по наличию выполняемых ими специфических функций. По реализуемым функциям защиты информации все подсистемы разделены на четыре основных группы:

1) подсистемы обеспечения дискретного управления доступом (Discretionary Access Control, DAC);
2) подсистемы обеспечения повторного использования объектов (Object Reuse, OR);
3) подсистемы обеспечения идентификации и аутентификации субъектов (Identification & Authentication, I&A) и
4) подсистемы контроля (Audit, AUD).

Эти категории образуют базис классификации компьютерных подсистем безопасности.

Для оценивания каждой функции защиты на основании сформулированных в отдельном документе интерпретаций требований TCSEC для подсистем определены соответствующие группы рейтингов. Подсистемы, которые оцениваются на наличие более чем одной функции, получают отдельный рейтинг для каждой оцениваемой функции (см. рис. 4).

Как уже отмечалось, подсистемы реализуют лишь отдельные функции безопасности и поэтому не могут соответствовать всем требованиям класса безопасности C1 и выше, определенным в TCSEC. Поэтому принято относить системы, использующие для защиты подсистемы безопасности, к классу D — классу "незащищенных" систем. Это и понятно, поскольку использование одной или нескольких подсистем в одной системе никоим образом не может гарантировать, что система полностью защищена. Именно это служит причиной появления в возможных рейтингах безопасности для подсистем буквы D (после наклонной черты).

Однако с учетом того, что использование подсистем все же в определенной степени повышает защищенность системы, в классе D для подсистем выделены три градации — D1, D2 и D3.

Класс D1 сопоставляется подсистемам, соответствующим интерпретации требований, предъявляемых к системам класса C1 в TCSEC.

Аналогично, класс D2 сопоставляется подсистемам, соответствующим интерпретации требований, предъявляемых к системам класса C2 в TCSEC.

Наконец, класс D3 зарезервирован для подсистем DAC и AUD, соответствующих интерпретациям

Функция подсистемы	Класс безопасности	Возможные рейтинги
Дискретное управление доступом	DAC/D	DAC/D1 DAC/D2 DAC/D3
Повторное использование объекта	OR/D	OR/D2
Идентификация и аутентификация	I&A/D	I&A/D1 I&A/D2
Контроль	AUD/D	AUD/D2 AUD/D3

Рис. 4. Возможные рейтинги подсистем

функциональных требований класса B3 в TCSEC.

Подобно классам в TCSEC, классы подсистем иерархически упорядочены. Это, в частности, означает, что подсистема, отнесенная к высшему классу, должна соответствовать всем функциональным требованиям, предъявляемым ко всем низшим классам, и всем дополнительным требованиям, предъявляемым к данному классу.

Кроме функциональных требований, подсистема должна соответствовать определенным для каждого класса требованиям доверия к подсистеме и требованиям к документации на подсистему.

Еще раз подчеркиваем, что подсистемы не могут даже в совокупности составлять некоторого завершенного решения безопасности компьютерной системы. Иными словами, не существует общего алгоритма получения рейтинга безопасности результирующей системы на основании рейтингов имеющихся в ней подсистем безопасности. Чтобы получить какую-либо оценку безопасности в таком случае, нужно проводить специальное оценивание системы в целом со всеми имеющимися в ней подсистемами. Рейтинг подсистемы, входящей в состав системы, не является, таким образом, какой-либо самостоятельной составляющей рейтинга системы в целом.

Указанные выше функции безопасности находятся во взаимной зависимости. Например, для выполнения подсистемой функции DAC необходимо наличие так называемых средств поддержки, выполняющих функции I&A и AUD. Некоторые функции вообще зависят циклически: например, DAC зависит от I&A, но и I&A зависит от DAC. Такая зависимость создает определенные трудности в выделении группы требований, относящихся только к данной функции. Кроме того, по этой же причине существует фундаментальная проблема в оценивании подсистем, поскольку существует явная зависимость между средствами безопасности, реализующими ту или иную функцию, которая ограничивает "независимость" включения подсистемы в среду системы. А именно: для эффективного включения подсистемы в среду защищаемой системы необходимо, чтобы в этой среде существовали средства, реализующие функции поддержки для данной подсистемы. Выполняться это условие может двумя способами: либо в самой подсистеме существуют средства, выполняющие функции поддержки, либо подсистема обеспечивает интерфейсы для использования соответствующих средств, входящих в состав защищаемой системы или других подсистем. Сложность оценивания во втором случае состоит в том, что для выяснения качества реализации подсистемы должны быть проверены все требуемые интерфейсы, тогда как сама логика исполнения подсистемой функций защиты не должна допускать такого рода проверки (поскольку в противном случае допускается возможность внешнего вмешательства в работу подсистемы, что сводит на нет все усилия по обеспечению безопасности). Именно по этим соображениям более приемлемым решением считается реализация поддерживающих функций в составе самой подсистемы.

Рассмотрим указанные классы подсистем.

1. Подсистемы класса DAC/D1 должны определять и контролировать в системах ADP доступ поименованных пользователей (субъектов) к поименованным объектам системы (например, файлам, программам, устройствам).

Дискретное управление доступом является одним из основополагающих компонентов воплощения в различных продуктах (как системах, так и подсистемах) политики безопасности, состоящей, по определению, в присвоении субъектам и объектам идентификаторов и фиксации набора правил, используемых для определения, имеет ли данный субъект авторизацию, достаточную для получения к данному объекту данного типа доступа.



К средствам DAC относятся: механизмы ограничения доступа к объектам; механизмы определения авторизации для субъектов по отношению к объектам; механизмы

для установления, просмотра и отмены привилегий доступа. Механизм реализации (например, списки управления доступом, профили пользователей, дескрипторы и т.д.) должен позволять пользователю определять и контролировать вид использования (тип доступа) управляемых данным пользователем объектов другими поименованными пользователями либо поименованными группами пользователей, либо и то, и другое.

Конкретно для класса DAC/D1 требуется, чтобы подсистема управляла доступом на уровне поименованных групп пользователей. Принятие решений по разрешению или запрещению доступа осуществляется подсистемой на основании доступных ей идентификаторов субъектов и объектов. Это предполагает, что подсистема должна иметь встроенные средства I&A либо обеспечивать интерфейс к аналогичным средствам, имеющимся в защищаемой системе или других подсистемах.

Особо следует обратить внимание на требование, согласно которому подсистема должна предоставлять отдельному пользователю возможность указывать, каким образом другой пользователь или группы пользователей могут получить доступ к объекту, владельцем которого он (первый пользователь) является. Это может быть осуществлено либо предоставлением пользователю возможности изменять в определенных пределах авторизации других пользователей, либо сопоставлять информацию о правах доступа к объекту других пользователей.

Важным моментом является также то, что подсистема должна опосредовать все попытки доступа, предпринимаемые в системе. Иными словами, между самой системой и пользователями всегда должна стоять подсистема.

2. Подсистемы класса DAC/D2, помимо всех требований, предъявляемых к подсистемам класса DAC/D1, должны удовлетворять дополнительным требованиям.

Прежде всего это касается избирательности средств идентификации пользователей: для подсистем данного класса каждый пользователь должен быть уникально идентифицирован (т.е. идентификация группы пользователей уже недостаточна).

Далее, подсистема класса DAC/D2 должна обеспечивать взаимодействие со средствами контроля (AUD), имеющимися в защищаемой системе или других подсистемах, а также генерировать соответствующие данные контроля, передаваемые указанным средствам.

От подсистем требуется также ограничивать возможность пользователей расширять права доступа к объектам, причем как в отношении прав доступа других пользователей, так и в отношении прав доступа самого пользователя.

Кроме этого, от подсистем данного класса требуется так называемая *установка защиты по умолчанию*. Смысл этого механизма состоит в том, чтобы подсистема отказывала в предоставлении доступа к данному объекту **всем пользователям** в случае, если авторизованный пользователь явно не определил права доступа других пользователей.

3. Подсистемы класса DAC/D3 характерны тем, что, помимо всего уже перечисленного, они должны обеспечивать возможность пользователям определять для каждого объекта список лиц, имеющих доступ к данному объекту. При этом должна существовать возможность указывать в списке режим(-ы) доступа, разрешенный(-ые) каждому пользователю в отношении данного объекта. Более того, пользователь должен иметь возможность создавать для каждого своего объекта список лиц, которым запрещен какой-либо доступ к данному объекту.

Единственным приемлемым механизмом, способным обеспечить выполнение данного требования, являются списки управления доступом (Access Control List; ACL).

4. Подсистемы класса OR/D2 предназначены для предотвращения попыток незаконного получения критичной информации, остатки которой могли сохраниться в некоторых объектах памяти, ранее использованных и освобожденных другим пользователем.

Чтобы понять приведенную формулировку, следует вспомнить, что когда, например, операционная система сообщает вам, что по вашему желанию некоторый фай удален, это вовсе не означает, что содержавшаяся в данном файле информация уничтожена в прямом смысле слова. Удаление файла означает только то, что система пометила блоки памяти, ранее составлявшие файл, специальным флажком, говорящим о том, что данный блок не входит в состав какого-либо файла и может быть использован для размещения в нем информации. Но ведь та информация, которая ранее была записана в данном блоке, никуда не исчезает! То есть, если непосредственно прочитать сам блок (а это вполне можно сделать), то можно беспрепятственно получить "удаленную" информа-



цию! Описанная стратегия "удаления" по ряду вполне объективных причин используется в подавляющем большинстве компьютерных систем.

Функция OR собственно и состоит в устранении возможности считать остатки информации. Делается это либо затиранием остатков информации какой-либо бессмыслицей (в простейшем случае — просто блоком сплошных нулей или единиц), либо — более тонко — за счет отказа любому пользователю в возможности прочитать "свободный" блок до тех пор, пока пользователь полностью не заполнил его собственной информацией. Официально требования к подсистемам класса OR/D2 формулируются следующим образом:

"До выделения из пула неиспользуемых объектов памяти для любого субъекта должны быть проверены все авторизации, касающиеся информации, содержащейся в данном объекте. Никакая информация, включая шифрованное ее представление, созданная в процессе предшествующих действий субъекта, не должна быть доступна какому-либо субъекту, получившему доступ к объекту, ранее возвращенному системе."

Приложение А

В таблицах 1 и 2 представлена хронология принятия законов по компьютерной безопасности в США и некоторых стандартов по закрытию информации в компьютерных системах.

Таблица 1

Основные законодательные и исполнительные акты США, касающиеся проблем компьютерной безопасности

Год	Исполнительная область	Законодательная область	Основные отчеты
Первая мировая война	Министерство обороны [1]		
1934		Закон о связи	
1952	Создано NSA [2]		
1965		Закон Брукса (Brooks)	
1976			Отчет Сенату о федеральных разведывательных функциях
1977	NBS устанавливает DES как стандарт на территории США		Отчет корпорации MITRE по безопасности в линиях связи
1978		Закон о противодействии иностранным разведкам [3]	
1979	Политика в области защиты правительственных линий связи PD/NSC 24 [4]		Отчет корпорации RAND по компьютерной безопасности
1980			"Белая книга" NTIA
1981	Исполнительный закон 12333 [5]		
1984	Политика в области защиты правительственных компьютеров и линий связи NSDD 145		
1985		Слушание в Белом доме о политике в области компьютерной безопасности	
1986	Политика в области защиты критичной информации	Закон о компьютерной безопасности HR 145	
1987		Слушание в Белом доме по закону HR 145	Доклад Белому дому по компьютерной безопасности

[1] Министерство обороны занималось обеспечением безопасности передачи дипломатических сообщений и сигналов разведывательных операций.
[2] Агентство Национальной безопасности (АНБ) (National Security Agency, NSA) было создано в соответствии с меморандумом президента.
[3] Этот закон устанавливает стандарты и процедуры при использовании электронной аппаратуры в правительственных учреждениях.
[4] Это директива президента и Совета национальной безопасности.
[5] Этот закон включает требования АНБ по обеспечению безопасности данных при их передаче.

Таблица 2
Хронология разработки основных алгоритмов и стандартов обеспечения безопасности информации

Стандарт	Разработчики/год	Использование
Стандарт шифрования данных (Data Encryption Standard - DES)	NBS/1977	Правительство США (в ЭВМ и линиях связи); все чаще используется в частном секторе
DES режимы для операций (DES Modes of Operation)	NBS/1980	Правительство США (управление ключами при передаче сообщений, телефонных разговорах)
Система подтверждения ключа (Key Notarization System)	NBS/1980	Правительство США (при подтверждении личности и при приеме секретных сообщений); также используется в банках
Директива по использованию DES (Guidelines for Implementing the DES)	NBS/1981	Правительство США
Идентификация данных на ЭВМ (Computer Data Authentication)	NBS/1985	Правительство США (идентификационный код для безопасности данных в системах и сетях); также используется в частном секторе
Стандарт на использование паролей (Password Usage Standard)	NBS/1985	Правительство США (указываются десять факторов безопасности для систем с парольной защитой)
Общие требования по безопасности для аппаратуры, использующей DES (General Security for Equipment Using DES)	GSA/1982	Правительство США (физическая и электрическая безопасность устройств, использующих DES)
Требования DES к безопасности на физическом уровне передачи данных (Interoperability and Security Requirements of the DES in the Physical Layer of Data Communication)	GSA/1983	Правительство США
Алгоритм шифрования данных (Data Encryption Algorithm - DEA)	ANSI/1981 X3.92	Промышленность США
Стандарт шифрования в линиях связи (Data Link Encryption Standard)	ANSI/1983 X3.105	Промышленность США
DEA-режимы для операций (DEA Modes of Operation)	ANSI/1983 X3.106	Промышленность США
Идентификация сообщений финансового свойства (Financial Institution Message Authentication)	ANSI/1983 X9.9	Банки (идентификация сообщений) и промышленность
Персональный идентификационный номер (ПИН). Управление и безопасность (Personal Identification Number (PIN). Management and Security)	ANSI/1982 X9.8	Банки
Управление ключами финансовых сообщений (Financial Institution Key Management)	ANSI/1985 X9.17	Банки
Управление и использование ПИН (Management and Use of PINs)	ABA/1979	Банки
Защита ПИН при обмене (Protection of PINs in Interchange)	ABA/1979	Банки
Документ по стандарту управления ключами (Key Management Standard Doc.)	ABA/1980	Банки
Алгоритм шифрования данных (DEA-1)	ISO/1986	Международная версия DES
DEA-1-режимы для операций (Modes of Operation of DEA-1)	ISO/DIS 8372	
Алгоритм шифрования в линиях связи (Data Link Enciphering Standard)	ISO/DIS 9160	Международный стандарт, версия ANSI X9.105
Алгоритм и системы шифрования общими ключами (Public Key Encryption Algorithm and Systems)	ISO/DP 9307	

Таблица 2 (окончание)

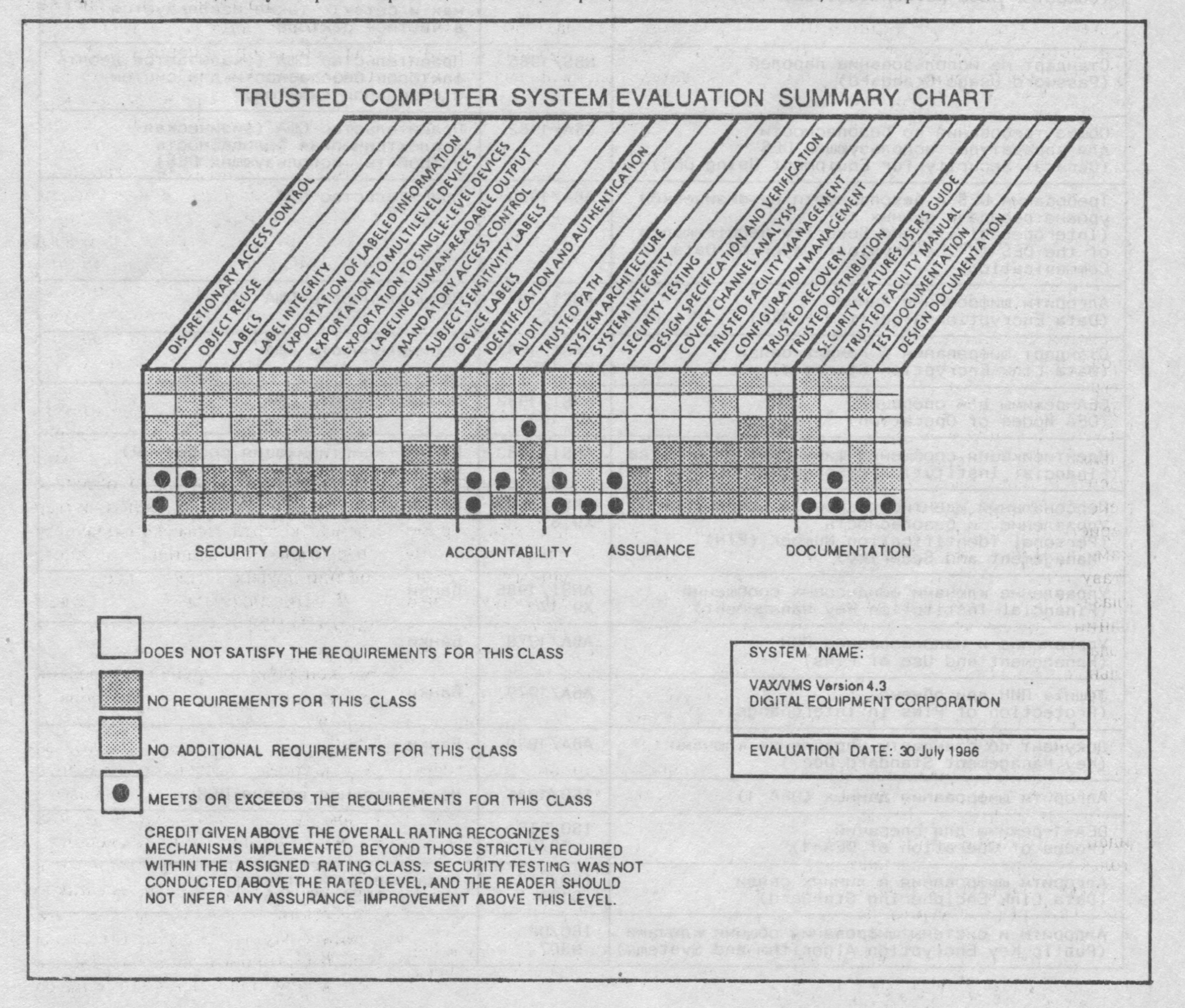
Идентификация сообщений (Message Identification)	ISO/DIS 8730	Международный стандарт идентификации сообщений; часть 1 включает спецификацию алгоритма DEA-1, часть 2 включает спецификацию алгоритма MAA
Банки: управление ключами (Banking: Key Management)	ISO/DIS 8732	Международный стандарт для банков

Примечания:

ABA — American Bankers Association — Ассоциация американских банкиров (США).
ANSI — American National Standard Institute — Американский национальный институт стандартов, филиал ISO.
GSA — General Services Administration — Администрация общих служб (США).
ISO — International Organization for Standardization — Международная организация по стандартизации.
NBS — National Bureau of Standards — Национальное Бюро Стандартов (США).

Приложение Б

Так реально выглядит суммарная карта оценивания операционной системы VAX/VMS версии 4.3.



Напоминаем, что в качестве объекта памяти в данном случае могут фигурировать не только блоки файлов, но и различного рода буферы, кадры страниц памяти, секторы магнитных дисков, зоны магнитных лент, регистры памяти и так далее.

5. Подсистемы класса I&A/D1 обеспечивают выполнение еще одной из основополагающих функций защиты — идентификацию-аутентификацию пользователей. На основании информации, получаемой в результате выполнения этих функций, строится вся политика безопасности.

Идентификация пользователя заключается в том, что пользователь при выполнении каких-либо действий должен себя назвать, вернее указать идентификатор, присвоенный данному пользователю в данной системе.

Следует отметить, что идентификация не обязательно состоит во вводе некоторого имени с клавиатуры. Для идентификации пользователя могут применяться (и успешно применяются!) специальные устройства, способные идентифицировать пользователя по некоторым его физическим характеристикам, например, по отпечаткам пальцев, спектральному составу голоса или даже по сетчатке глаза. Кроме того, для идентификации применяются различного рода магнитные карточки, специальные ключи и т.д.

После идентификации выполняется аутентификация полученного идентификатора, то есть проверяется содержательность указанного идентификатора для данной системы. В самом деле, требование идентификации заключается только во вводе идентификатора. А если пользователь в ответ на запрос введет какое-нибудь слово, первым пришедшее на ум? Если бы идентификация не дополнялась аутентификацией, то сама иден-



тификация теряла бы всякий смысл. Часто аутентификация пользователя основана на запросе ввода пользователем пароля либо ответа на некоторые тестовые вопросы. В любом случае для проведения аутентификации пользователь должен выполнить некоторые явные действия. Следует отметить, что механизмы, используемые для выполнения аутентификации, должны быть устойчивы к подлогу, подбору или подделке*.

Подсистемы класса I&A/D1 должны требовать от пользователя идентификации до начала выполнения этим пользователем любых действий, реализуемых средствами защищаемой системы. При этом для систем данного класса допускается применение пользователем группового идентификатора. Кроме того, подсистема должна обеспечивать защиту данных аутентификации, гарантирующую, что эти данные не смогут быть получены каким-либо неавторизованным пользователем.

Подсистема I&A должна быть надежно связана с защищаемой системой для передачи последней аутентифицированной идентификации пользователя. Этот интерфейс должен быть надежно защищен от любых попыток внешнего воздействия.

6. Подсистемы класса I&A/D2 должны обеспечивать уникальную идентификацию каждого пользователя. Это означает, в частности, что должно обеспечиваться гарантированное распознавание подсистемой каждого члена каждой группы пользователей, а также распознавание специфических пользователей системы (например, администратора системы, оператора системы, офицера безопасности и т.д.).

Подсистема данного класса должна обеспечивать ведение специального *контрольного журнала*, в котором должны фиксироваться все события, связанные с идентификацией-аутентификацией. В запись контрольного журнала должны входить: источник запроса (например, идентификатор терминала), дата и время поступления запроса, поступивший идентификатор, тип события, отметка об успехе или неуспехе события. Следует отметить, что допускается либо реализация этого требования (ведение контроля) встроенными средствами самой подсистемы, либо обеспечение подсистемой надежного интерфейса к средствам контроля, существующим в защищаемой системе или других подсистемах.

7. Подсистемы класса AUD/D2 служат для обеспечения возможности ведения в системе контроля за функционированием самой системы и контроля за работой с системой пользователей. Собственно контроль (Audit) заключается в выделении, накоплении в едином месте (так называемом *следе контроля* — audit trail), защищенном хранении и предоставлении по требованию авторизованного (для выдачи такого требования) пользователя специальных данных о различных типах событий, происходящих в системе и так или иначе влияющих на со-

* Здесь имеет смысл упомянуть еще такой процесс, как авторизация пользователей. Под авторизацией пользователя понимается определение прав данного пользователя на доступ к различным объектам системы на основании введенного пользователем идентификатора.

стояние безопасности данной компьютерной системы.

След контроля не тождествен **контрольному журналу**; **контрольный журнал** — всего лишь частный вид реализации **следа контроля**. По определению **след контроля** — это совокупность записей, содержащих требуемую контрольную информацию, представленную в установленном формате. Таким образом, след контроля может быть реализован в виде специальной области памяти, доступ к которой может быть осуществлен исключительно средствами подсистемы контроля, либо в виде отдельного устройства, опять же защищенного от незаконного доступа со стороны неавторизованных для этого пользователей.

Контроль проводится в основном для достижения двух целей:

a) для отслеживания текущего состояния безопасности в защищаемой системе, своевременного обнаружения возможности нарушения безопасности и предупреждения об этом лиц, отвечающих за безопасность системы;

b) для обеспечения возможности обратной трассировки (по данным контроля) совершенного нарушения безопасности с целью обнаружения причин данного нарушения и установления степени ответственности причастных к нарушению лиц.

Подсистема контроля должна обеспечивать возможность сбора и хранения требуемых данных о всех событиях, происходящих в любом месте защищаемой системы и имеющих отношение к безопасности.

Естественно, что для нормального функционирования подсистема контроля должна тесно взаимодействовать со средствами I&A или иметь встроенные средства этого типа.

Подсистема контроля должна обеспечивать для администратора системы возможность выборочного контроля за действиями одного или нескольких пользователей, то есть возможность выделения из всей совокупности накопленных данных контроля записей, относящихся к действиям конкретного лица, либо накопления только данных контроля, относящихся к действиям указанных системным администратором лиц.

Конкретно подсистемы контроля данного класса должны обеспечивать создание и ведение *контрольного журнала.* Если сама система по каким-либо причинам не может обеспечить получение и сбор требуемых данных контроля для каких-либо мест защищаемой системы, подсистема должна иметь надежный интерфейс к средствам системы или других подсистем, обеспечивающих требуемые действия. Во всех остальных случаях (когда компоненты системы могут создавать данные контроля и передавать их подсистеме) подсистема должна надежно принимать передаваемые ей данные и записывать их в *следе контроля*.

Напоминаем, что механизм контроля, параметры контроля, среда хранения данных контроля и сами данные контроля должны быть надежно защищены от попыток неавторизованного доступа. Лица, авторизованные для доступа к данным контроля (например, администратор системы или офицер безопасности), должны получать доступ только через саму подсистему контроля. Для обеспечения выполнения этого требования подсистема контроля должна тесно взаимодействовать со средствами DAC либо иметь встроенные средства этого типа. Допускается и рекомендуется физическая защита данных контроля.

8. Подсистемы класса AUD/D3, помимо всех требований, предъявляемых к подсистемам класса AUD/D2, должны обеспечивать выполнение ряда дополнительных требований, а именно должны отслеживать возникновение или накопление в системе событий, угрожающих ее безопасности, и немедленно оповещать администратора системы при превышении некоторого установленного предела. В случае, когда течение или накопление угрожающих событий продолжается, подсистемы данного класса должны самостоятельно предпринимать некоторые шаги, нацеленные на блокировку или полное разрушение угрожающих безопасности системы действий.

Фирмы — разработчики программного обеспечения весьма заинтересованы в получении и повышении официальных рейтингов безопасности для своей продукции, поскольку наличие рейтинга безопасности является необходимым условием для получения фирмой значительных государственных и военных заказов, а также служит хорошей рекламой для продукции фирмы.

С NCSC сотрудничают фирмы AT&T, Boeing, Clyde Digital Systems, Computer Security, Data General, Digital Equipment, Enigma Logic, Fisher-Inns, Gemini, Gould, Harris, Hewlett-Packard, IBM, Key Concepts, Micronix, Pike Greek, Prime, Sperry (UNISYS), Verdix, Wang, Xerox.

Одной из первых фирм, поддержавших программу NCSC, была фирма DEC.

В заключение хотелось бы выразить большую надежду на то, что в недалеком будущем мы тоже сможем подходить к разработке, оцениванию, тиражированию и распространению программно-аппаратных средств отечественного и совместного производства столь же серьезно и основательно, как это делают нынешние законодатели компьютерных мод — американцы.

И.Моисеенков

Использованы материалы:

Information Systems Security. Products and Services Catalogue, July, 1987.

Computer Security Subsystem. Interpretation of the Trusted Computer System Evaluation Criteria, 1988, NCSC-TG-009, version-1.

Графические форматы данных

Формат GIF

Формат GIF (Graphics Interchange Format) был предложен еще в 1987 году американской фирмой Compuserve как протокол для обмена графическими данными через MailBox. Несмотря на достаточно широкое распространение GIF-файлов, структура самого формата известна не очень многим программистам и пользователям персональных компьютеров. Публикуя эту статью, мы в какой-то мере постараемся восполнить этот пробел.

Вообще говоря, формат GIF достаточно сильно отличается от других графических форматов файлов. Одной из его особенностей является возможность размещения в одном GIF-файле нескольких изображений. Кроме того, размеры этих изображений могут составлять 16 000x16 000 точек и состоять из 256 цветов палитры в 16 миллионов цветов. Благодаря тому, что таблица цветов (RGB) имеет размер 24 бита, формат GIF является практически аппаратно-независимым и позволяет достаточно точно воспроизводить оригинальное изображение. Примерная структура GIF-файла приведена на рис. 1.

Заголовок GIF-файла
- GIF-сигнатура
- Блок Определения Экрана
- Глобальная Таблица Цветов (необязательно)
- Блок Расширения (необязательно)

Блок Данных
- Блок Определения Изображения
- Локальная Таблица Цветов (необязательно)
- Данные

Конец GIF-файла
- GIF-терминатор = ";"

Рис. 1

Каждый такой файл состоит из Заголовка, собственно данных изображения (Блок Данных) и Признака Конца файла. Так как формат GIF позволяет разместить в одном файле несколько изображений, то независимые друг от друга Блоки Данных могут размещаться последовательно.

Заголовок GIF-файла может иметь переменную длину, так как состоит из обязательных и необязательных (optional) компонентов. Первым обязательным элементом Заголовка является сигнатура — буквы GIF, после которых следует номер версии, обычно "87a". Следующим обязательным компонентом Заголовка является Блок Определения Экрана — это блок глобальных данных, характеризующих устройство, на котором было получено изображение (рис. 2).

Блок Определения Экрана

Смещение, байт	Содержание
6	Горизонтальное разрешение экрана в пикселах
8	Вертикальное разрешение экрана в пикселах
10	Флаг Разрешения
11	Цвет фона
12	00h

Рис. 2

Два 16-разрядных слова определяют соответственно горизонтальное и вертикальное разрешение экрана. Эти значения могут быть больше, чем действительные размеры изображения. Поэтому пользователь сам должен позаботиться о подходящем размещении изображения на экране своего монитора. Пятый байт Блока Определения Экрана называется Флагом Разрешения (рис. 3). Младшие три бита этого байта (0, 1, 2) определяют число бит на один пиксел. Иначе говоря, они определяют максимальное число цветов в изображении. Счет начинается с нуля, поэтому значение 3 будет означать 4 бита на пиксел, что соответствует 16

цветам. Поскольку через три бита можно записать наибольшее число 7, то максимально возможное количество цветов будет 256. Четвертый бит в настоящее время не используется и обычно равен нулю. Биты 5, 6 и 7 определяют количество бит, используемых для таблицы цветов RGB. Стандартным значением является число 7, и поскольку счет по-прежнему начинается с нуля (7+1 = 8), каждой цветовой составляющей (красный, синий, зеленый) соответствует один байт, а всей таблице — 24 бита. Однако формат GIF не различает эти цветовые составляющие и предполагает наличие таблицы постоянной ширины. Восьмой бит Флага Разрешения указывает на наличие (или отсутствие) Глобальной Таблицы Цветов. Если она существует, то непосредственно используется для определения экрана, иначе программа чтения GIF-файла должна иметь наготове собственную таблицу цветов. Это связано с тем, что изображения не обязательно должны сопровождаться Локальными Таблицами Цветов. Если же Глобальная Таблица Цветов все-таки существует, то для каждого изображаемого цвета в ней ставятся в соответствие значения красного, зеленого и синего цветов. Значения RGB для определенных цветов расположены последовательно, что может быть описано, например, следующей структурой данных

```
RGBValues: Array [0...MAXCOLOR]
OF ARRAY[0...2] OF BYTE
```

Отдельные значения RGB лежат между 0 и 255, причем ширина RGB-таблицы соответствует значению из байта Флаг Разрешения Блока Определения Экрана. Наряду с Глобальной Таблицей Цветов для данных изображения может быть определена и Локальная Таблица Цветов, которая, вообще говоря, имеет некоторые преимущества. Структура Локальной Таблицы Цветов полностью соответствует Глобальной.

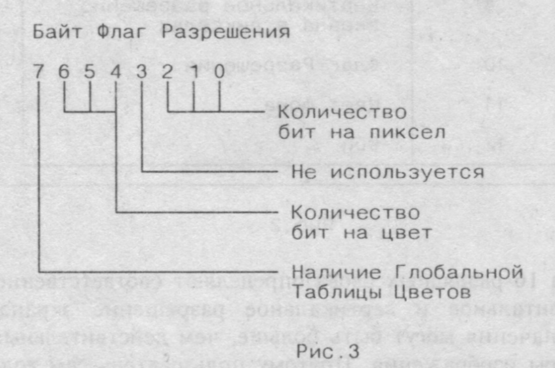


Рис.3

Блок Расширения является необязательным элементом GIF-файла и расположен обычно между Глобальной Таблицей Цветов и Блоком Данных изображения.

Программа чтения GIF-файла, вообще говоря, может этот блок просто игнорировать. Первым байтом Блока Расширения всегда является символ "!" (шестнадцатиричный код 21h), а последним — терминатор 00h.

Как уже отмечалось, для каждого изображения, включенного в GIF-файл, существует свой Блок Данных. Этот Блок состоит из трех элементов: Определение Изображения, Локальная Таблица Цветов (необязательно) и сами данные о пикселах. Элемент Определения Экрана содержит важные сведения о самом изображении (рис. 4).

Блок Определения Изображения

Смещение, байт	Содержание
0	Разделитель =","
1	Горизонтальное смещение изображения
3	Вертикальное смещение изображения
5	Ширина изображения
7	Высота изображения
9	Байт Флага Изображения

Рис.4

По сути — это некий Заголовок изображения, который начинается с обязательного символа "," (код 2Ch). После этого следуют два 16-разрядных слова, определяющих соответственно горизонтальное и вертикальное смещение верхнего левого угла изображения. Ширина и высота изображения в пикселах содержится в последующих двух словах. Причем как ширина, так и высота изображения должны быть либо меньше, либо равны соответствующим размерам экрана, представленным в Заголовке GIF-файла. Десятым байтом Определения Экрана является Флаг Изображения, закодированный побитно (рис. 5). Нижние три бита представляют количество бит на пиксел для данного изображения. Самый старший (бит 7) определяет, используется ли Глобальная Таблица Цветов (бит 7 = 0) либо речь идет о Локальной Таблице. Бит 6 определяет, расположены ли данные изображения последовательно или используется режим чередования (Interlaced Mode).

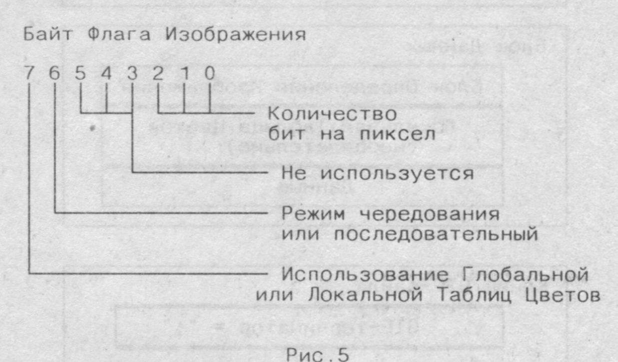


Рис.5

В последовательном режиме строки изображения располагаются одна за другой, в то время как при использовании режима чередования строки запоминаются в четыре приема. Сначала запоминается каждая восьмая строка (0, 8, 16 ...), затем каждая восьмая строка, начиная с пятой (5, 12, 20 ...). После этого сохраняется каждая четвертая строка начиная с третьей (3, 7, 11 ...) и, наконец, все недостающие. Режим чередования очень удобен при передаче изображения с использованием телефонных линий, так как даже при медленной передаче на принимающей стороне изображение появляется достаточно быстро, и при передаче очередной порции строк качество его улучшается.

Собственно данные изображения хранятся за Определением Изображения или за Локальной Таблицей Цветов. Эти данные закодированы достаточно эффективно, но и достаточно сложно. В качестве метода сжатия выбран LZW-алгоритм, используемый, например, в таких популярных программах-архиваторах как PKARC и PKZIP. Однако упаковка данных в формате GIF имеет ряд особенностей. Вот несколько из них.

Формат GIF, например, определяет специальный Clear-код (Adaptives Reset) для обнуления LZW-таблицы. Причем этот код принимает не фиксированное, а одно из двух максимальных значений. Для стандартного изображения EGA с 16 цветами это может быть число 16. После того как такое число читается или пишется, все LZW-таблицы обнуляются. Для того, чтобы каким-то образом определить окончание данных изображения, GIF-формат использует так называемый код EOI (End Of Information). Этот код имеет значение Clear-код плюс один.

Формат PCX

Когда в 80-е годы фирмой ZSoft для сохранения изображений, полученных при помощи пакета PC PaintBrush, был предложен формат PCX, мало кто мог предположить, что в дальнейшем этот формат получит столь широкое распространение.

В настоящее время многие профессиональные пакеты позволяют использовать формат PCX для создания файлов, хранящих графическое изображение. Хотя, в принципе, PCX-изображение может быть получено с использованием практически любого мало-мальски приличного графического адаптера (например, CGA), но тесная связь формата PCX с пакетом-родителем PaintBrush, безусловно, наложила свой отпечаток. Дело в том, что первоначально использовать сохраненное изображение в PCX-файлах предполагалось только на том оборудовании, на котором оно было получено, что и нашло свое отражение в структуре формата. К тому же о появлении новых, более мощных графических адаптеров разработчики в то время, очевидно, не думали.

Любой PCX-файл начинается с заголовка размером в 128 байт, из которых до настоящего времени используется только 70 (табл. 1). Версии PCX-файлов 2.8 и 3.0 (байт версии равен или 2, или 5) имеют дополнительный блок цветовой палитры, который физически расположен в конце PCX-файла, сразу за данными изображения. Как уже отмечалось, разработчики PCX-формата не предполагали использование для него VGA-адаптера с 256 цветами, поэтому зарезервировали для красного, синего и зеленого цветов только 48 байт (по 16 на каждый). Поэтому блок дополнительной цветовой палитры расположен несколько необычно и занимает при этом 768 байт для VGA-адаптера с 256 цветами. В этом случае PCX-файл обозначает цвета в диапазоне от 0 до 255. Как известно, EGA-адаптер

Таблица 1

Смещение	Обозначение	Пояснения
0	Creator	Всегда – 10 (ZSoft)
1	Version	Версия PCX: 0 – версия 2.5 2 – версия 2.8 с цветовой палитрой 3 – версия 2.8 без цветовой палитры 5 – версия 3.0 с цветовой палитрой
2	Encoding	Кодирование RunLength – 1
3	Bits	Число бит на пиксел и цветовую плоскость: CGA (низкого разрешения) и VGA с 256 цветами – 2 бита все остальное – 1 бит.
4	Xmin, Ymin, Xmax, Ymax	Логические координаты изображения
12	HRes	Горизонтальное разрешение
14	VRes	Вертикальное разрешение
16	Palette	Цветовая палитра 16x3 байта 16x(R,G,B)
64	VMode	Для внутреннего использования в PaintBrush
65	Planes	Число цветовых плоскостей: EGA/VGA – 4 CGA/Hercules – 1
66	BytePerLine	Число байт на строку только для одной цветовой плоскости
68	PaletteInfo	Интерпретация цветности: цвет (также и черно-белый) – 1 ступень серого – 2
70		Зарезервировано

может корректно обрабатывать значения цветов только от 0 до 63. Поэтому для него используется четыре диапазона цветов, а для получения правильного значения цвета используется деление на 64 (табл. 2).

Данные изображения для PCX-формата построчно сжимаются с использованием Run-Length-кодирования, так как оно хорошо подходит для сжатия повторяющихся байтов. Ведь, как правило, графические изображения довольно часто имеют одноцветные области, которые описываются одинаковыми байтами. Этот факт и использует PCX-формат, в котором последовательность повторяющихся байтов заменяется только двумя байтами. Причем сжатие происходит лишь в том случае, если имеется не меньше трех одинаковых байтов. Отсюда следует, что в PCX-файле данные об изображении могут быть представлены в виде как закодированных, так и незакодированных байтов. Признаком кодирования является единичное значение двух старших бит в байте (биты 6 и 7 равны 1). В этом случае шесть оставшихся бит байта (биты 0-5) указывают на число повторений следующего байта, несущего информацию о цвете. При кодировании следует обратить внимание на тот факт, что максимальное число повторений одного байта в PCX-файле не может быть больше 63, так как в качестве счетчика используются только 6 разрядов.

Таблица 2

Значение цвета для VGA	Диапазоны цветов для EGA
0 – 63	0
64 – 127	1
128 – 191	2
192 – 255	3

Не следует забывать и о том, что изменение строки и цветовой плоскости (bit plane) может происходить в пределах одного повторения. Таким образом, программа, читающая PCX-файл, должна найти выход, как в данном случае интерпретировать данные. Можно, конечно, использовать переменную из заголовка файла, определяющую количество байтов на строку (см. рис. 1). Эта переменная всегда содержит количество байтов, которые необходимы для полной цветовой плоскости одной строки экрана. Здесь следует обратить внимание на то, что PCX-изображение может оказаться больше, чем экран. При этом, безусловно, на экран должны быть скопированы лишь те байты, которые физически смогут быть отображены.

А. Борзенко

dBASE IV для начинающих

Глава 3. Работа с данными

Начиная с этой главы мы переходим к более подробному изучению dBASE. Мы уже многое знаем. Многое нам уже известно. Как работает меню системы, как создать файл базы данных, как ввести данные.

В этой главе мы рассмотрим следующие темы:
- просмотр введенных записей;
- редактирование уже введенных записей;
- использование в dBASE информации разных типов;
- ввод новой информации;
- модификация структуры базы;
- уничтожение данных;
- ввод и корректировка текстовой информации;
- копирование данных из файла в файл.

3.1. Просмотр данных

Теперь мы научимся тому, как извлекать введенные записи из базы данных и просматривать на экране дисплея.

Итак, перед нами управляющий центр. Выберите из панели Data файл TELEF. Для этого подведите курсор к этому файлу и нажмите Enter.

Появится рамка, которая предоставит вам три выбора (см. рис. 3.1.):
- Use file — сделать файл базы данных активным (использовать файл);
Modify structure/order — изменить поля в структуре или порядок данных (модифицировать структуру/порядок);
- Display data — просмотреть содержание файла (показать данные). Что нам и требуется.

Выберите опцию Display data.

Теперь на экране вы видите первую запись файла TELEF, приведенную на рис. 3.2. Поля в файле имеют те же имена, типы и размер, которые вы задали при создании структуры файла TELEF. Для просмотра каждой следующей записи нажимается клавиша PgDn.

Но вообще-то очень неудобно просматривать файл по одной записи, а если база будет большая, то это вдобавок и утомительно. dBASE предоставляет другую возможность для просмотра файла.

```
      Catalog   Tools   Exit                                       4:52:07 pm
                           dBASE IV CONTROL CENTER
                      CATALOG: C:\DBASE\UNTITLED.CAT
        Data       Queries       Forms        Reports        Labels    Applications
     <create>     <create>     <create>      <create>      <create>     <create>
                  Use file    Modify structure/order    Display data
                        Press ENTER to select or ESC to cancel

   File:          TELEF.DBF
   Description:
   Select option and press ENTER, or press first letter of desired option
```

Рис. 3.1

Продолжение. Начало в №1'92 и №2'92

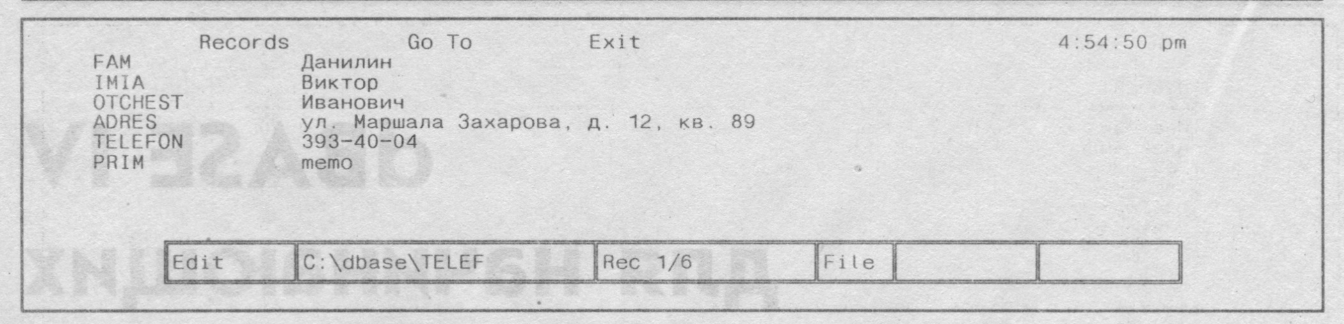


Рис. 3.2

```
Records   Fields   Go To   Exit                                4:58:09 pm
```

FAM	IMIA	OTCHEST	ADRES
Данилин	Виктор	Иванович	ул. Маршала Захарова, д. 12,
Большакова	Татьяна	Александровна	ул. 13-я Парковая, д. 13, кв
Брун	Владимир	Ефимович	Староконюшенный пер., д. 4,
Ушакова	Елена	Алексеевна	Армянский пер., д. 10, кв. 3
Кожухова	Ольга	Альбертовна	ул. Сталеваров, д. 10, кв. 1
Чесаков	Давид	Моисеевич	ул. Тверская, д. 10, кв. 50

```
Browse | C:\dbase\TELEF | Rec 1/6 | File
View and edit fields
```

Рис. 3.3

```
Records   Fields   Go To   Exit                                4:59:40 pm
```

ADRES	TELEFON	PRIM
ул. Маршала Захарова, д. 12, кв. 89	393-40-04	memo
ул. 13-я Парковая, д.13, кв. 6	464-35-43	memo
Староконюшенный пер., д. 4, кв. 11	415-23-44	memo
Армянский пер., д. 10, кв. 3	385-15-12	memo
ул. Сталеваров, д. 10, кв. 15	404-35-04	memo
ул. Тверская, д. 10, кв. 50	285-56-82	memo

```
Browse | C:\dbase\TELEF | Rec 1/6 | File
View and edit fields
```

Рис. 3.4

Нажмите клавишу F2. Перед вами возникнет табличка, где каждая запись расположена в отдельной строке (рис. 3.3.). В dBASE такой табличный формат, как вы видите на рисунке, назван Экраном Просмотра (Browse), поскольку вы можете использовать его для быстрого просмотра всех записей базы.

Но что это такое? При ближайшем рассмотрении файла TELEF в режиме BROWSE вы обнаруживаете, что не все поля, записанные при проектировании этой базы, показаны на экране. Не волнуйтесь, они просто не уместились на экран. Достаточно нажать на клавишу Tab (табуляция). Вы увидите, как курсор с первого поля перепрыгнул на второе. Так и продолжайте. Когда курсор подойдет к границе экрана, и вы нажмете еще раз клавишу Tab, первое поле базы как бы уйдет за левую границу экрана, но зато то поле, которое мы не видели раньше, появится у правой границы (см. рис. 3.4.). Таким образом мы можем просмотреть и все остальные поля. Перемещение в обратную сторону выполняется клавишами Shift-Tab.

3.2. Редактирование данных

Рано или поздно вам все равно понадобится обновить данные в вашей базе. Записи могут быть изме-

```
    Records    Fields    Go To    Exit                          5:02:10 pm
 ┌───────────────────┬──────────────────────────────────────┬───────────┬───┐
 │OTCHEST            │ADRES                                 │TELEFON    │P  │
 ├───────────────────┼──────────────────────────────────────┼───────────┼───┤
 │Иванович           │ул. Маршала Захарова, д. 12, кв. 89   │393-40-04  │m  │
 │Александровна      │ул. 13-я Парковая, д. 13, кв. 6       │464-35-43  │m  │
 │Ефимович           │Староконюшенный пер., д. 4, кв. 11    │415-23-44  │m  │
 │Алексеевна         │Армянский пер,. д. 10, кв. 3          │385-15-12  │m  │
 │Альбертовна        │ул. Ткацкая, д. 2, кв. 50. 15         │404-35-04  │m  │
 │Моисеевич          │ул. Тверская, д. 10, кв. 50           │285-56-82  │m  │
 └───────────────────┴──────────────────────────────────────┴───────────┴───┘
 │Browse  │C:\dbase\TELEF        │Rec 5/6        │File   │    │          │
                          View and edit fields
```

Рис. 3.5

нены, когда ваши знакомые будут менять место жительства, телефон и т.п.

В dBASE изменение данных проводится очень просто. Вы можете изменять введенные данные как в режиме Browse, так и в режиме Edit. Мы будем исправлять данные в режиме Browse.

Допустим, ваша знакомая Кожухова Ольга Альбертовна переехала с улицы Сталеваров на Ткацкую (д. 2, кв. 50). Следовательно, вам необходимо внести изменение поле ADRES шестой записи.

Используя клавиши Tab и "стрелка вниз", подведите курсор к полю ADRES шестой записи.

Наберите на клавиатуре новый адрес.

Обратите внимание на то, что часть старого адреса все равно осталась (рис. 3.5.). Уберите их с помощью клавиши Del (Уничтожить).

Использование клавиши Ins. Вы можете вводить текст как в режиме Вставки (Ins), так и без него.

Когда режим Вставки выключен, символы, которые вы набираете с клавиатуры, как бы забивают старый текст. А если вы работаете в режиме Вставки, то введенные символы отодвигают старый текст.

По умолчанию режим Вставки в dBASE выключен, но его легко включить, нажав на клавишу Ins.

3.3. Добавление новых записей

Вы можете вводить информацию, используя любой из двух режимов: Просмотра (Browse) или Редактирования (Edit).

```
 1 FAM       Пальмихина
 2 IMIA      Елена
 3 OTCHEST   Юрьевна
 4 ADRES     пр. Вернадского, д.45, кв. 123
 5 TELEFON   135-54-13
 6 PRIM
```

Рис. 3.6

Давайте посмотрим, как используются опции меню режима Редактирования для ввода новых данных.

1. Из подменю Records (Записи) выберите опцию Add new records (Добавить новые записи).

   Эта опция подготавливает для ввода новой информации пустую запись. Курсор появляется на первом поле записи. Теперь вы можете ввести данные в каждое поле.

2. Напечатаем вашу информацию. Допустим, вы хотите записать координаты вашей новой знакомой. Данные о ней представлены на рис. 3.6.

Действуйте так же, как и при вводе информации при первоначальном заполнении нашей базы. По окончании не забудьте нажать комбинацию клавиш Ctrl-End для записи информации и выхода в управляющий центр.

3.4. Ввод и редактирование текстовой информации

Открытие текстовых полей

Для открытия текстового поля подведите курсор к этому полю и нажмите клавиши Ctrl-Home.

На экране появится рабочая область текстового поля (см. рис. 3.7.).

Ввод и редактирование информации в текстовых полях

Информация в текстовых полях вводится и исправляется точно так же, как и при работе на пишущей машинке. Для перехода на новую строку используется клавиша Enter, для исправления ошибок — клавиши Backspace и Del.

Сохранение текста

Когда вы закончите редактировать или вводить информацию в текстовое поле, нажмите на комбинацию клавиш Ctrl-End. Текстовое поле закроется, и курсор появится в отметке текстового поля.

1. Нажмите Ctrl-End для закрытия и сохранения данных в поле PRIM.

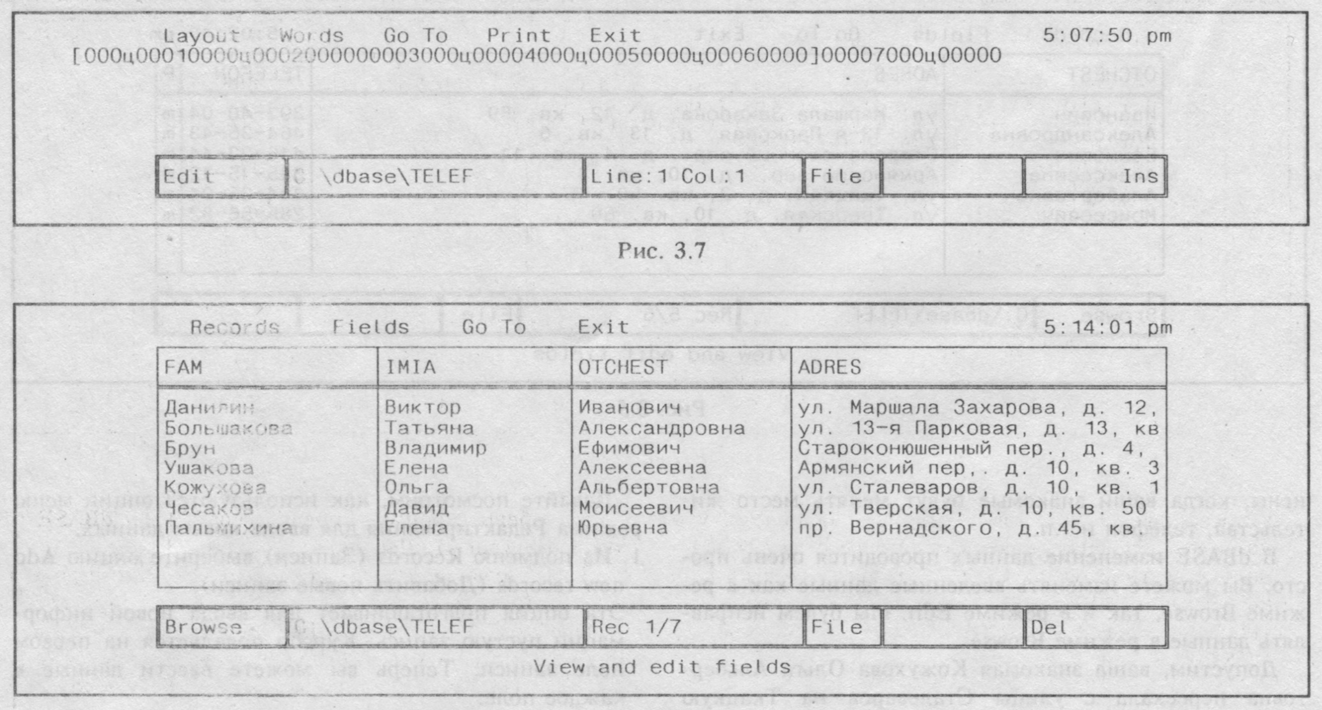


Рис. 3.7

Рис. 3.8

2. Выберите опцию Exit из меню Exit (Выход) для возврата в Управляющий Центр.

3.5. Удаление информации

Записи уничтожаются в два этапа:

1) запись, которую вы хотите удалить, помечается;

2) вы даете команду удаления.

Допустим, вы хотите удалить первую запись нашей базы.

1. Войдите в режим Просмотра базы.
2. Нажмите клавишу F10 для выхода в меню (курсор должен стоять на записи, которую вы хотите удалить).
3. Выберите опцию Mark record for deletion (Отметить запись для удаления) из меню Records (Записи).

В полосе состояния справа появится индикатор Del, показывающий, что данная запись помечена на удаление (см. рис. 3.8).

То, что запись помечена на удаление, еще не говорит о том, что она удалена. Ничего пока еще не потеряно. Мы можем всегда ее восстановить, для этого:

1. Встаньте опять на эту запись (посмотрите, что индикатор Del горит).
2. Нажмите F10 для выхода в меню Records, выберите опцию Clear deletion mark (Очистить отметку на удаление).
3. Нажмите Enter

Индикатор Del исчезнет.

Есть более простой и легкий способ отметки записи на удаление и возврата в старое положение. Для этого, встав курсором на запись, которую вы хотите удалить, нажмите комбинацию клавиш Ctrl-U. В полосе состояния появится индикатор Del. Если вы нажмете эту комбинацию еще раз, индикатор удаления исчезнет.

Но мы можем после того, как пометим запись на удаление, убрать ее из нашей базы совсем.

Для этого в режиме проектирования базы данных выберите из меню Organize (организовать) опцию Erase marked records (Уничтожить помеченные записи). dBASE IV спросит, действительно ли вы хотите убрать эти записи из файла. Переведите курсор к Yes (Да) и нажмите Enter.Выберите опцию Save changes and exit (Сохранить изменения и выйти) из меню Exit (выход).

3.6. Копирование баз данных

Ввод каждой записи с клавиатуры занимает много времени, если вам надо ввести много записей. Однако если у вас есть уже готовые данные, скомпонованные в базу, dBASE может скопировать их в новую базу, сохранив вам часы работы.

Копирование в другие базы данных

Во избежание случайной потери информации следует делать резервные копии файлов. Есть три пути копирования информации:

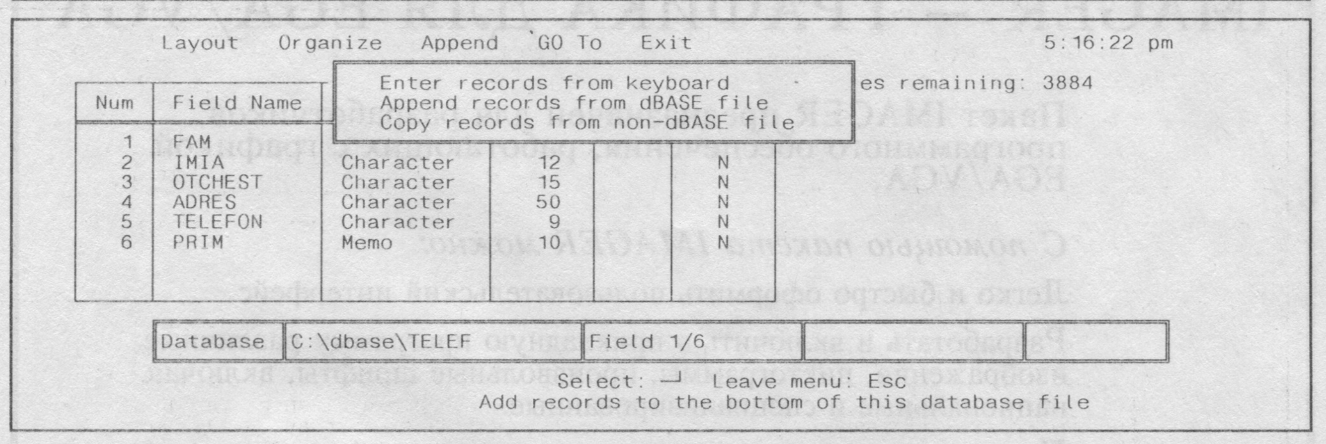


Рис.3.9

- записать файл под новым именем, используя утилиты DOS, содержащиеся в меню Средства (Tools);
- использовать команду COPY TO, введя ее в командной строке dBASE;
- использовать команду COPY операционной системы DOS.

Передача данных из другого файла

Вы можете также добавить данные из одного файла dBASE в другой, для этого:

1) выберите файл, в который вы хотите скопировать данные, и войдите в режим проектирования базы данных;

2) выберите опцию Append records from dBASE file из меню Append (Добавить). См. рис. 3.9;

3) из появившегося списка выберите соответствующий файл.

Необходимо помнить, копируя данные из одного файла в другой, что они должны иметь структуры записей, содержащие одинаковые поля. Только эти поля будут копироваться.

А.Иванов

(Продолжение в одном из следующих номеров)

Новая телефонная карточка MCI

MCI выпустила новую телефонную карточку. Она объединяет в себе предоставление доступа к обычной телефонной связи, голосовой почте и справочным службам (новости, погода, спортивная и биржевая информация). Карточка выдается бесплатно, многие услуги тоже бесплатны, а за остальные вносится фиксированная плата. Те, кто пользуется программами, предоставляющими скидки (типа "Друзья и Семья"), могут получить свои 20% скидки при пользовании новой карточкой, а в качестве развития таких программ пользователи Sprint или AT&T, которые сдадут свои старые карточки и сделают MCI своим оператором "по умолчанию", получат возможность пользоваться некоторыми услугами бесплатно.

The Teleputing Hotline, January 13, 1991.

AT&T образует совместное предприятие на Украине для модернизации украинской телефонной сети. Это первое с момента создания СНГ крупное соглашение между государством, входившими ранее в состав СССР, и большой американской компанией. AT&T сообщила, что вкладывает в предприятие "десятки миллионов" долларов, в основном в виде 13 наисовременнейших цифровых коммутаторов 5ESS, которые начнут устанавливаться этой весной. Еще 12 будут изготовлены на Украине. Партнеры увеличат практически в два раза число местных телефонных номеров — с 7 до 13 миллионов. AT&T будет владеть 39% предприятия, нидерландская компания PTT Telecom — 10%, а владельцем 51% будет Украина. Будет также проложено 60 прямых линий для связи между Украиной и остальным миром, вместо использования 150 линий, идущих через Москву.

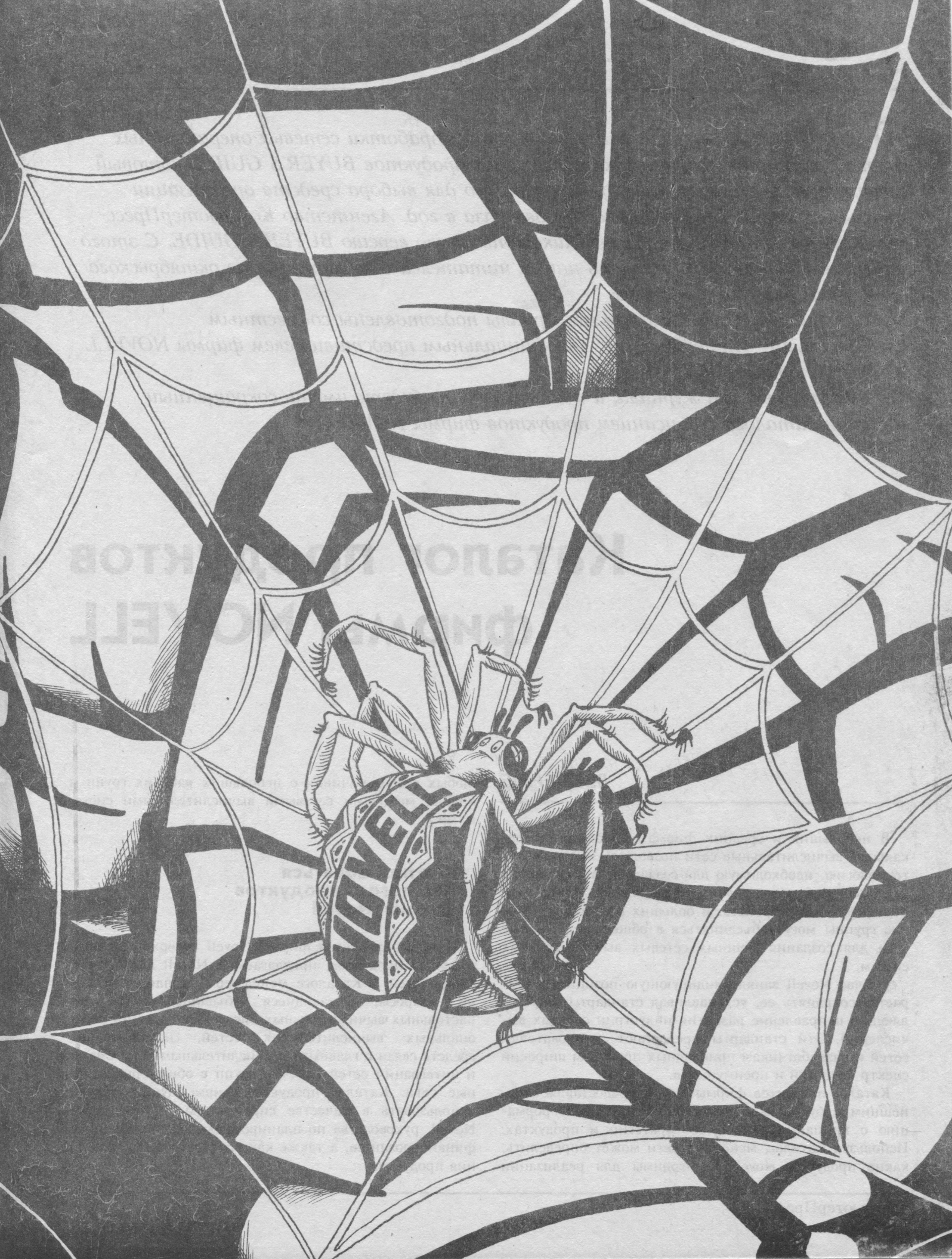
The Teleputing Hotline, January 20, 1991.

Кирилл Чащин сообщает Newsbytes, что практически с любого телефона в Москве в субботу и воскресенье, а также ночью, можно напрямую позвонить за границу. Однако цены увеличились в три раза. Звонок в Европу стоит 18 рублей в минуту, в США — 36 рублей и в Австралию — 42 рубля. По оценке Чащина, сейчас 1 рубль примерно равен 1-му центу. (Тарифы на передачу электронной почты по Relcom, которая является крупнейшей сетью в СНГ, поднялись в 4 раза.) Конгресс бирж, новая политическая организация биржевиков, заявил, что планирует обеспечить быструю передачу в частные руки Министерства связи, несмотря на возражения Президента России Бориса Ельцина.

The Teleputing Hotline, January 20, 1991.

Электронный бюллетень Canada Remote стал первой системой, предлагающей доступ к сети Usenet. Usenet дает BBS доступ к всемирной сети UUCP, объединяющей 12 миллионов пользователей со всего мира, работающих в основном в операционной системе UNIX. Canada Remote сообщила, что планирует предложить своим абонентам доступ к своим вычислительным ресурсам под управлением системы UNIX, к передаче файлов по Internet и доступ к архивным материалам.

The Teleputing Hotline, January 20, 1991.


NOVELL

Фирма NOVELL, мировой лидер в области разработки сетевых операционных систем, регулярно выпускает каталог своих продуктов BUYER'S GUIDE, который содержит полную информацию, необходимую для выбора средств организации локальных сетей. Каталог выходит два раза в год. Агентство КомпьютерПресс планирует в конце 1992 года опубликовать новую версию BUYER'S GUIDE. С этого номера мы начинаем знакомить наших читателей с выдержками из октябрьского (1991 года) выпуска каталога.
К публикации на русском языке материалы подготовлены совместным предприятием "ИнтерПроКом" — официальным представителем фирмы NOVELL на территории СНГ.
Сохраняйте выпуски журнала, и в конце года вы будете иметь сокращенный вариант каталога с описанием продуктов фирмы NOVELL.

Каталог продуктов фирмы NOVELL

ВВЕДЕНИЕ

В небольших и средних фирмах и компаниях локальные вычислительные сети позволяют использовать технологию, необходимую для создания рабочих групп и совместного использования прикладных программ и периферийных устройств. В больших компаниях рабочие группы могут объединяться в общекорпоративные сети для создания мощных сетевых вычислительных систем.

Сейчас Novell заняла лидирующую позицию и старается сохранять ее, устанавливая стандарты, определяющие направление развития индустрии сетевых вычислений. Эти стандарты предлагают пользователям сетей и разработчикам прикладных программ широкий спектр функций и преимуществ.

Каталог продуктов фирмы Novell предоставляет нынешним и будущим пользователям полную информацию о компании Novell, ее стратегии и продуктах. Используя Каталог, менеджер сети может определить, какие продукты Novell необходимы для реализации любых сетей, начиная с небольших рабочих групп и кончая мощными сетевыми вычислительными системами.

Как пользоваться Каталогом продуктов фирмы Novell

Каталог продуктов фирмы Novell содержит полный перечень продуктов, предлагаемых Novell для организации сетей. В Каталоге менеджер сети найдет ответы на вопросы, касающиеся объединения в сеть настольных вычислительных систем, реализации региональных вычислительных сетей, использования средств связи с главными вычислительными системами и интеграции сетей рабочих групп в общекорпоративные сети. Каталог продуктов фирмы Novell можно использовать в качестве справочника по продуктам Novell, руководства по планированию, эксплуатации и финансированию, а также как справочник для сравнения продуктов.

Каталог продуктов фирмы Novell состоит из пяти частей, содержащих:
- описание стратегии и структуры корпорации Novell;
- общее описание продуктов Novell;
- детальное описание продуктов Novell;
- общее описание методов поддержки продуктов и обучения пользователей;
- общее описание методов поддержки и обучения.

Описание каждого продукта Novell содержит следующую информацию:
- краткое определение продукта: название, его назначение и основные преимущества;
- характеристики продукта: характеристики и описание функциональных возможностей;
- необходимое аппаратное обеспечение: все аппаратные средства, необходимые для работы продукта;
- необходимое программное обеспечение: все необходимые дополнительные программные средства, например, конкретные версии DOS или NetWare;
- опции: все дополнительные продукты, о наличии которых должен знать пользователь;
- спецификации: технические спецификации продукта;
- информация для заказа: номер изделия Novell.

КОРПОРАЦИЯ NOVELL И ЕЕ СТРАТЕГИЯ

История компании

Novell — компания, разрабатывающая программное обеспечение сетевых систем: первоначально — локальных вычислительных сетей, состоящих только из персональных компьютеров, а сегодня — и более крупных сетей, объединяющих все вычислительные средства предприятия, включая большие и миниЭВМ.

Уже в 1983 году Novell осознала, что развитие микропроцессорной технологии приведет к созданию более мощных настольных ЭВМ, способных активно работать в сетях, включающих крупные центральные ЭВМ. Novell работала над тем, чтобы понимание этой тенденции стало достоянием всей компьютерной индустрии и заказчиков фирмы. Novell достигла успехов во многих областях. Сегодня фирма Novell не только удовлетворяет существующие потребности своих заказчиков, но и закладывает основу, на которой они смогут осуществить интеграцию всей информации, циркулирующей в организации.

Novell всегда ставила перед собой задачу ускорения развития индустрии сетевых вычислений и обеспечения лидирующих позиций в этом секторе компьютерного рынка. За восемь лет штат Novell увеличился до 2400 человек, работающих по всему миру. За этот период существенно увеличился объем продаж, достигнув в 1990 году 497,5 млн. долларов. В 1985 году он составлял лишь 55,3 млн. долларов.

Фактически Novell создала новый стандарт на совместное использование информации настольными ЭВМ, серверами сети, управляющими доступом к данным в ней и теперь еще большими и миниЭВМ.

Продукты NetWare

Программное обеспечение NetWare оптимизировано для управления, совместного использования, преобразования и синхронизации данных в сетевой среде. Операционная система NetWare определяет возможности сервера сети, управляющего совместным использованием подсистем связи, обработки файлов, печати, баз данных и передачи сообщений. Она позволяет организовать совместное использование этих функций и прикладных программ компьютерами различных типов, объединенными в одну сеть.

Novell создала сбалансированный набор сетевых продуктов, позволяющий удовлетворить самые разнообразные потребности, начиная с самых маленьких рабочих групп персональных компьютеров и систем внутри отделов и кончая крупнейшими общекорпоративными системами.

NetWare — это сетевая операционная система, получившая широкое распространение в промышленности, позволяющая менеджерам небольших предприятий и отделов беспрепятственно объединять в сети имеющиеся у них компьютеры. Расширенные возможности, предоставляемые семейством продуктов NetWare фирмы Novell, позволяют менеджерам информационных систем удовлетворять потребности средних и крупных компаний в интеграции информации.

Стратегия фирмы Novell, применяемая ею при разработке продуктов, дала заказчикам гибкую сетевую архитектуру. Обеспечивая заказчиков модулями для построения систем, Novell предоставляет им большую свободу при создании сетей на основе продуктов Novell и других поставщиков. При этом обеспечивается эффективное использование вычислительных ресурсов, какими бы сложными и неоднородными они ни были.

Усилия Novell в области создания продуктов наиболее полно отражены в операционной системе NetWare v.3.11. Выпущенная в 1989 году под названием NetWare 386, она позволяет создать открытую сетевую среду, интегрирующую ресурсы имеющихся серверов, настольных компьютеров, больших и миниЭВМ, и совместно использовать тысячи прикладных программ для компьютеров под операционными системами PC-DOS, Macintosh, UNIX или OS/2.

Открытая архитектура NetWare v.3.11 обеспечивает прозрачное, управляемое совместное использование данных компьютерами различных типов. Она также предлагает средства и возможности, традиционно ассоциирующиеся только со средами больших ЭВМ, такие как расширяемость, высокая производительность, надежность и устойчивость к сбоям. NetWare v.3.11 способна управлять дисковой памятью, сопоставимой с объемом памяти миниЭВМ, и обеспечивает более вы-

сокую производительность ввода-вывода информации при значительно более низких затратах.

Эта операционная система позволяет оптимизировать работу сетевого сервера и эффективно управлять сетью.

Среда NetWare v.3.11 содержит и другие продукты Novell, которые могут быть объединены с этой операционной системой. Дополнительный сервис включает поддержку пользователей Macintosh и UNIX, функции связи, архивирования и резервирования базы данных, а также функции управления сетью.

Создавая архитектуру NetWare 386 3.x, Novell не прекратила разработки продуктов семейства NetWare 2.x — наиболее популярной из сетевых операционных систем для создания рабочих групп. Последние достижения в этой области включены в новую версию данной операционной системы — NetWare 2.2, объединяющей лучшие качества ELS NetWare, Advanced NetWare и SFT NetWare в одном многофункциональном продукте, устанавливающем новый стандарт соотношения стоимость/производительность для сетей рабочих групп, и в то же время отличающемся простотой в инсталляции, использовании и управлении.

Сотрудничество с другими фирмами

Novell сотрудничает с другими поставщиками в самых различных областях. Независимые фирмы обычно поставляют прикладные программы семейства Portable NetWare, обеспечивающие функции NetWare. Благодаря этому внешние вычислительные среды могут функционировать в качестве серверов NetWare, поддерживая терминалы или настольные компьютеры. Такое решение позволяет прикладным программам настольных компьютеров напрямую взаимодействовать с серверами, обеспечивающими хранение больших объемов информации. Различные вычислительные системы становятся полностью совместимыми с серверами и рабочими группами на основе персональных компьютеров. Фирмы Altos Computer, Data General, Interactive Systems, NCR, Prime и Unisys поставляют продукты Portable NetWare с середины 1990 года. В конце 1991 года выпущены новые продукты Portable NetWare для компьютеров MIPS и HP 9000.

Novell поставляет программные инструментальные средства, облегчающие разработку связанных с NetWare прикладных программ. В рамках операционной системы NetWare v.3.11 эти инструменты позволяют разработчикам программного обеспечения поддерживать других поставщиков и конечных пользователей, создавая распределенные прикладные программы нового поколения, использующие преимущества многопроцессорных сетей. Разработчики создали и продолжают создавать для сетевой операционной системы NetWare v.3.11 широкий спектр распределенных прикладных программ. В их число входят программы баз данных, автоматизированного проектирования, электронной почты, функции связи, утилиты управления сетью, системные утилиты, такие как утилиты архивирования и резервирования, а также инструменты разработки программ и программного обеспечения рабочих групп.

Более 20 тысяч коммерческих и частных заказчиков используют продукты Btrieve и NetWare SQL фирмы Novell для создания баз данных. Простота программирования и широкое распространение NetWare способствовали созданию тысяч NetWare-совместимых утилит и прикладных программ. Сегодня на рынке имеется более 5 тысяч прикладных программ других поставщиков, использующих преимущества базовых протоколов NetWare.

Продукты NetWare поддерживают свыше 200 адаптеров ЛВС и более 100 систем дисковой памяти, систем резервирования информации и конфигураций серверов, одобренных Novell в рамках программы Independent Manufacturing Support Program.

Неоднородные функции и системы

Объединение Novell с фирмой Excelan в 1989 году позволило воспользоваться семилетним опытом ее специалистов в области создания систем с открытыми протоколами связи. Полученные дополнительные возможности позволили интегрировать в NetWare различные стандартные протоколы, включая AppleTalk File Protocol фирмы Apple, Network File System фирмы Sun Microsystems, работающий в UNIX, и стандартные протоколы открытых систем, такие как TCP/IP и OSI. Подобные протоколы, объединенные с операционной системой NetWare, — ключ к беспрепятственной и высокопроизводительной связи между ЭВМ различных типов.

Программные продукты обеспечения связи Novell включают NetWare for SAA, NetWare SNA Gateway, шлюзы X.25, эмуляторы терминалов 3270 и 5250, а также региональные мосты и программы маршрутизации в локальных и региональных вычислительных сетях.

Изделия Novell обеспечивают надежное объединение всех сервисных функций и ресурсов, необходимое в неоднородных сетях. Применяемые фирмой технологии, такие как функции имен и протоколы сетевого управления, укрепляют ее лидирующие позиции в области общекорпоративных сетевых вычислений.

Удовлетворение потребностей заказчиков

За последние 10 лет потребности заказчиков изменились. Можно сказать, что продукты NetWare как вызвали эти изменения, так и отразили их. В 1988 году Novell усовершенствовала возможности сетей рабочих групп и отделов, выпустив устойчивую к сбоям операционную систему SFT NetWare, продукт NetWare for VMS, связывающий персональные ком-

пьютеры с центральными ЭВМ DEC, и программу NetWare for Macintosh, позволяющую компьютерам Macintosh использовать информацию совместно с персональными компьютерами IBM.

Отвечая потребностям заказчиков 90-х годов, Novell заложила начало новой ориентации систем, выпустив NetWare v.3.11 и организовав сотрудничество с партнерами для создания продуктов Portable NetWare. Оба этих шага направлены на беспрепятственную интеграцию больших и миниЭВМ с рабочими станциями. Партнеры по программе Portable NetWare увеличивают присутствие продуктов Novell в средах более крупных вычислительных систем за счет продажи своих систем и маркетинга.

На международной арене Novell совершенствует обслуживание и поддержку, создавая свои местные представительства. Для укрепления своих позиций на международном рынке Novell осуществляет локализацию своих продуктов и их обслуживания.

Novell является партнером нескольких фирм, разделяя собственность с фирмой Softbank, дистрибьютором продуктов компьютерной промышленности, а также с такими производителями ЭВМ, как Canon, Fujitsu, NEC, Toshiba и Sony.

В 1990 году объем продаж за границей составил 40% от общего объема продаж. Сегодня это одно из наиболее быстро растущих направлений деятельности Novell. Дочерние предприятия Novell, расположенные в Канаде, Бельгии, Франции, Германии, Италии, Испании, Швейцарии, Великобритании, Австралии, Гонконге и Японии, сообщают о результатах в центр международных операций, находящийся в гор. Саннивейл (шт. Калифорния).

Лидерство на рынке

Один из лучших методов оценки потенциала компании в будущем — взгляд на ее достижения в прошлом. Диаграмма на рис. 1.1 иллюстрирует лидирующее положение Novell на рынке. На ней указано время появления новых технологий, разработанных Novell. Novell, сознавая ответственность как лидера, постоянно стремится к развитию индустрии в целом.

Эволюция рынка вычислительных систем

Зарождение сетевых вычислений произошло в начале 80-х годов на базе сетей персональных настольных компьютеров. Службы и данные, основанные на операционных системах настольных компьютеров, таких как PC-DOS, были изолированы и выделены для одного пользователя. Изоляция функций в автономных персональных компьютерах, наряду с таким преимуществом, как обеспечение пользователя "персональной" вычислительной системой, в то же время затрудняла совместное использование данных в защищенной среде. По мере роста парка установленных персональных компьютеров потребность в совместном использовании данных становилась все более актуальной. Эта потребность послужила толчком к переходу от автономных персональных компьютеров к рабочим группам персональных компьютеров, объединенным локальными вычислительными сетями.

На сервере основой ресурсов и данных становится уже сетевая операционная система, а не операционная система рабочей станции. Размещение ресурсов на сервере позволяет использовать информацию совместно, а функции управления и защиты, также базирующиеся на сервере, обеспечивают целостность и защиту данных. Хотя в сетях ресурсы перестают быть выделенными для конкретного настольного компьютера, ощущение персональной вычислительной системы сохраняется благодаря прозрачности, обеспечиваемой для каждого компьютера. Иными словами, совместно используемые ресурсы внешне не отличаются для пользователя от локальных ресурсов.

Простота совместного использования информации и ресурсов в среде рабочих групп привела к значительным изменениям в стратегии корпораций в области вычислительных систем. Сетевые серверы эффективно конкурируют при создании вычислительных систем отделов с большими и миниЭВМ, традиционно использовавшимися для этих целей. В результате пользователи все больше полагаются на серверы. Пе-

1981	● Апрель: Появление IBM PC на рынке
1983	● Январь: Novell первой выпускает программное обеспечение файлового сервера ● Май: Novell представляет концепцию независимости от среды передачи
1985	● Март: IBM выпускает DOS 3.1 (поддержка сетей) и NetBIOS ● Сентябрь: Novell - первая независимая сетевая компания, поддерживающая DOS 3.1 и NetBIOS
1986	● Январь: Novell выпускает первую ОС ЛВС, работающую в защищенном режиме, для процессора 80286. Novell обеспечивает службы маршрутизации, независимые от среды передачи ● Март: IBM выпускает сеть Token-Ring Novell - первая независимая сетевая компания, поддерживающая Token-Ring ● Сентябрь: Novell выпускает продукты ЛВС, впервые имеющие устойчивость к сбоям оборудования
1987	● Сентябрь: Novell представляет технологию Open Protocol Technology ● Октябрь: IBM выпускает OS/2 стандартной редакции
1988	● Сентябрь: Novell выпускает NetWare for VMS, NetWare Requester for OS/2 ● Ноябрь: IBM выпускает OS/2 расширенной редакции Novell Requester for OS/2 поддерживает расширенную редакцию ● Декабрь: Novell обеспечивает поддержку Macintosh с протоколом AFP
1989	● Февраль: Novell представляет Portable NetWare ● Май: Novell представляет NetWare Open Systems ● Сентябрь: Novell выпускает первую 32-х разрядную ОС ЛВС для процессора 80386
1990	● Январь: Novell выпускает сетевой монитор, работающий по протоколу SNMP ● Апрель: Первый стратегический партнер, NCR, поставляет Portable NetWare ● Июль: Novell выпускает сервер с архитектурой, впервые поддерживающей несколько протоколов и сред передач

Рис. 1.1. Достижения в технологии

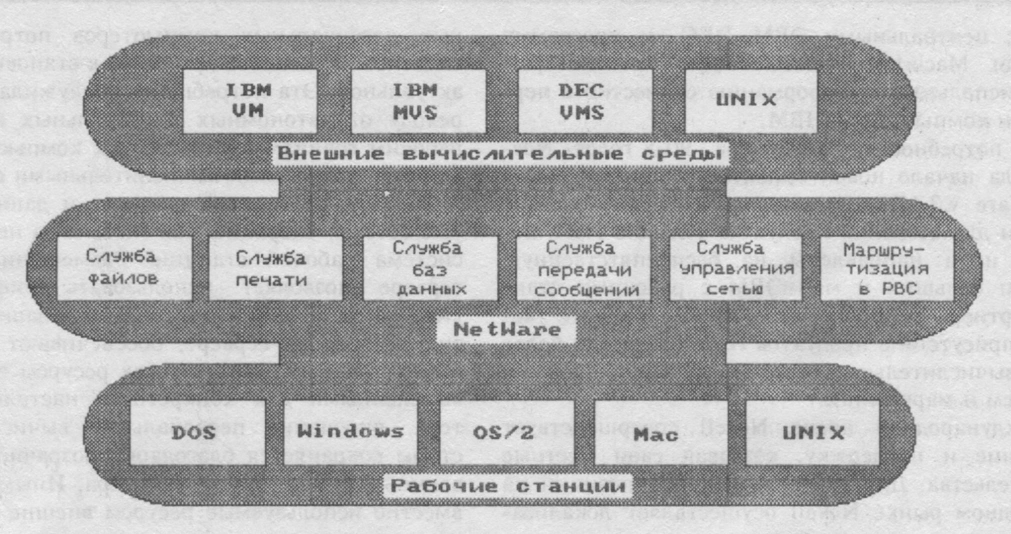


Рис. 1.2. Сетевая вычислительная система

реход от совместно используемых обрабатывающих систем к системам распределенной обработки происходит одновременно с изменением самой структуры бизнеса.

По мере того как бизнес становится все менее иерархичным, он требует все более гибких вычислительных систем, способных удовлетворять потребности организаций, а не жестких систем, потребности которых вынуждены удовлетворять сами организации. Высокоцентрализованная природа традиционных систем совместного использования информации на базе аппаратных и программных средств одного поставщика уже не отвечает запросам развивающегося бизнеса.

Эти тенденции ведут к исчерпанию резервов вычислительных систем на основе рабочих групп. По мере того, как специалисты по информационным системам приступают к реализации общекорпоративных систем, ограничения среды рабочих групп становятся все более очевидными. Появляется необходимость организации такой связи между миниЭВМ и большими ЭВМ, с одной стороны, и системами на основе рабочих групп, с другой, которая позволяла бы пользователям осуществлять доступ к ресурсам главных систем. При расположении ресурсов на сервере пользователям необходимо знать, где находится каждый ресурс, к которому они хотят осуществить доступ. Функции управления, "привязанные к серверу", также ограничивают свободу супервизора, осуществляющего управление общекорпоративной системой. Эти ограничения и послужили причиной перехода к технологии распределенной обработки данных — к сетевым вычислениям.

Сетевые вычисления

Типичное современное предприятие использует центральные системы, основанные на операционных системах VMS фирмы DEC, VM и MVS фирмы IBM, UNIX, а также серверы, подобные серверам, использующим NetWare. Рабочие станции могут использовать операционные системы DOS, Windows, OS/2, Macintosh и UNIX.

Сетевые вычисления обеспечивают интеграцию центральных и настольных систем в единую логическую сеть. Интеграция систем рабочих групп с системами на основе архитектур миниЭВМ и больших ЭВМ позволяет сократить капитальные вложения в эти системы и сделать их ресурсы доступными всем пользователям сети. Пользователи могут осуществлять доступ к данным, прикладным программам и вычислительным ресурсам этих ЭВМ так же легко, как и к локальному серверу. Каналы связи региональных вычислительных сетей позволяют объединять в одну сеть удаленные друг от друга системы, также обеспечивая прозрачный доступ к ресурсам.

Сетевые вычислительные системы такого типа создаются с одной целью: обеспечить прозрачное распределение информации и вычислительных функций между всеми пользователями организации, независимо от их местонахождения. В то время, как в системах рабочих групп ресурсы располагаются на определенном сервере, в рамках сетевых вычислений в роли платформы, обеспечивающей сетевые сервисные функции, выступает сама сеть. Распределяя функции по всей сети, сетевые вычисления делают эти функции доступными любому пользователю в организации, а не только расположенному близко к серверу. Этот тип систем показан на рис. 1.2.

Однако создание неоднородной сети — это только часть решений, обеспечиваемых сетевыми вычислениями. Системы "сетевых вычислений" могут рассматриваться не просто как сеть, состоящая из индивидуальных машин, но как представляющее самостоятельную ценность вычислительное устройство. Прикладные программы, использующие преимущества такого устройства, рассматривающие сеть как единую обра-

батывающую единицу, знаменуют собой конечную фазу сетевых вычислений. Эти прикладные программы, обычно называемые распределенными, требуют прозрачной межпрограммной связи и взаимодействия в среде, объединяющей различные ЭВМ, сетевые операционные системы и стандарты операционных систем рабочих станций.

NICA Интегрированная архитектура фирмы Novell

Интегрированная вычислительная архитектура фирмы Novell — Integrated Computing Architecture (NICA) — эта архитектура, определяющая построение всех продуктов Novell. NICA отражает подход Novell к предоставлению сетевых функций для интеграции прикладных программ в распределенной неоднородной среде. Используя архитектуру NICA, Novell стремится создать единое решение, основанное на концепции открытого программирования, которое позволило бы новым и существующим прикладным программам совместно использовать информацию и ресурсы системы независимо от того, какие это прикладные программы, где они размещаются и для вычислительных систем каких изготовителей предназначены.

Архитектура, используемая Novell для создания распределенных сетевых функций, позволяет объединять продукты и прикладные программы различных поставщиков в сложную сетевую вычислительную систему.

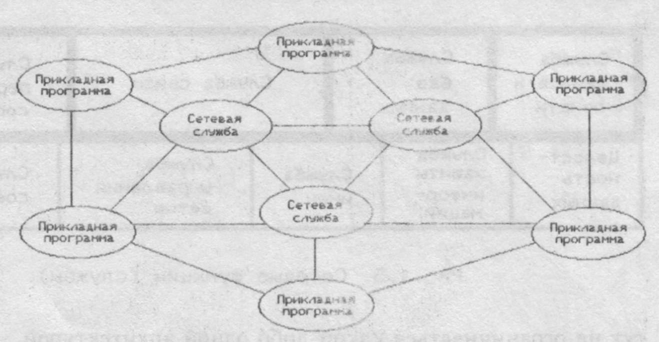
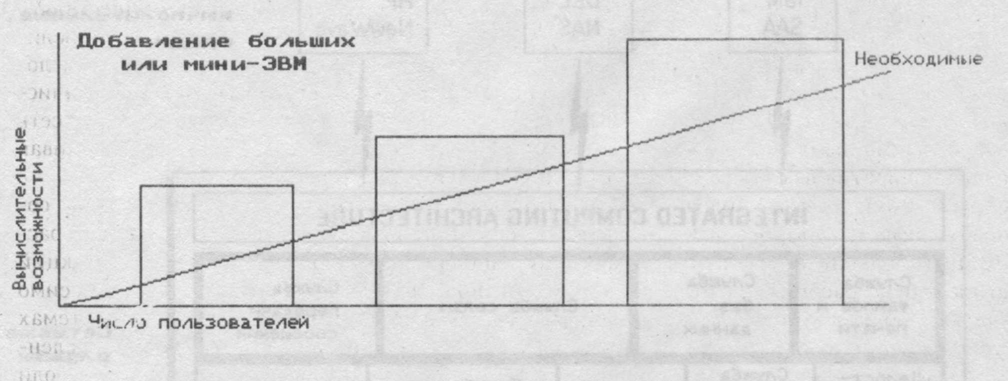


Рис. 1.3. Прикладные программы могут быть распределены между сетевыми функциями

Разработав архитектуру Integrated Computing Architecture, Novell тем самым признает тот факт, что вычислительные системы, использующиеся в современном бизнесе, разнородны, и приветствует интеграцию других архитектур в рамках NICA. Архитектура Novell Integrated Computing Architecture учитывает особенности других архитектур, определенных для распределения прикладных программ по сетям, таких как System Application Architecture (SAA) фирмы IBM, Network Application Support (NAS) фирмы DEC и NewWave Office фирмы Hewlett-Packard.

NICA является полезным дополнением этих прикладных архитектур, реализуя сетевые функции, позволяющие прикладным программам таких поставщиков, как IBM, DEC или HP, беспрепятственно выполняться в распределенной среде. На рис. 1.3 показано, каким образом осуществляется распределение прикладных программ и функций управления сетью в рамках архитектуры NICA.

Стратегия Novell в области поддержки прикладных программ направлена на создание сетевых средств, совершенствующих работу распределенных прикладных программ. Заказчики мо-



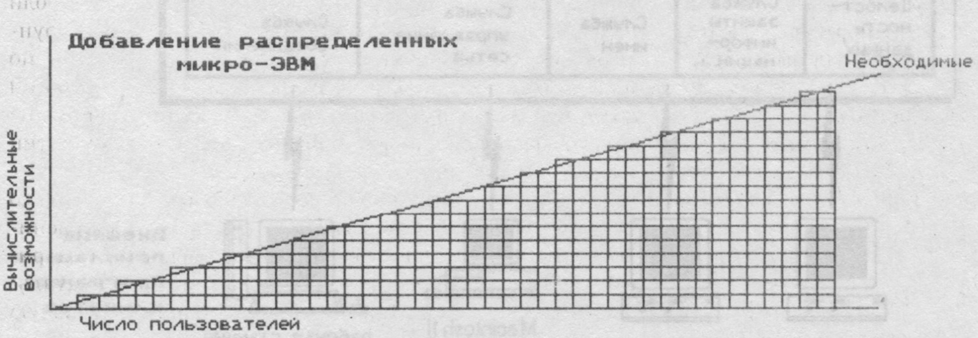


Рис. 1.4. Распределенные микроЭВМ обеспечивают постепенный рост вычислительных возможностей систем

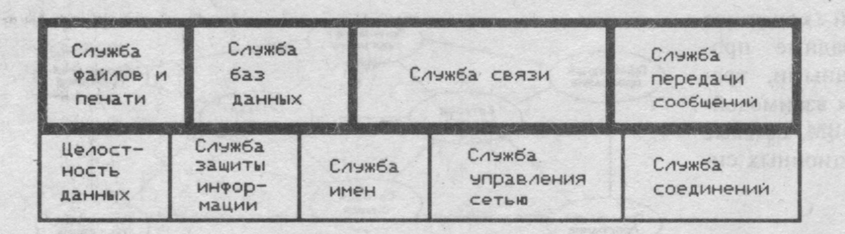


Рис.1.5. Сетевые функции (службы).

прикладных программ, работающих в средах операционных систем общего назначения, таких как DOS, Windows, OS/2, UNIX, Macintosh, MVS, VM или VMS, может быть реализовано при помощи сетевых функций специализированной сетевой операционной системы, например, такой как NetWare. На рис. 1.5 показаны сетевые функции, доступные в рамках архитектуры NICA.

В рамках NICA прикладные программы могут выполняться как на рабочих станциях пользователей (клиентах), так и на сервере. Прикладные программы требуют взаимодействия с пользователем, тогда как прикладные программы сервера выполняются в автономном режиме, не требуя вмешательства человека.

Кроме сетевых функций, необходимых для прикладных программ клиента, архитектура Integrated Computing Architecture фирмы Novell обеспечивает функции взаимодействия с прикладными программами, разработанными для архитектур SAA фирмы IBM, NAS фирмы DEC и NewWave office фирмы Hewlett-Packard. Рис. 1.6 иллюстрирует интеграцию этих различных прикладных архитектур.

Другим преимуществом распределенной сетевой операционной системы является возможность переноса

гут не ограничиваться какой-либо одной архитектурой. В рамках архитектуры NICA любые прикладные программы, удовлетворяющие спецификациям TCP/IP, могут быть распределены по сети. Возможности NICA выходят за рамки распределенной обработки, предоставляемой миниЭВМ и большими ЭВМ. Эта архитектура позволяет интегрировать распределенные прикладные программы и процессы различных типов. Заказчики, таким образом, могут сделать капитальные вложения в имеющиеся прикладные программы и добавлять по мере необходимости новые прикладные программы.

Одно из главных преимуществ распределенной сетевой операционной системы над миниЭВМ и большими ЭВМ — это возможность гибкого, постепенного, экономически обоснованного развития системы. Использование мини- и больших ЭВМ часто ставит менеджеров информационных систем перед необходимостью покупки излишних ресурсов. Как показано на рис. 1.4, использование распределенных микроЭВМ позволяет менеджерам информационных систем более эффективно планировать рост вычислительных возможностей.

Большинство существующих прикладных программ были разработаны для использования в операционных системах общего назначения. Сетевые функции позволяют совместно использовать эти прикладные программы и распределять их между несколькими процессорами в локальных или региональных вычислительных сетях. Распределение и совместное использование

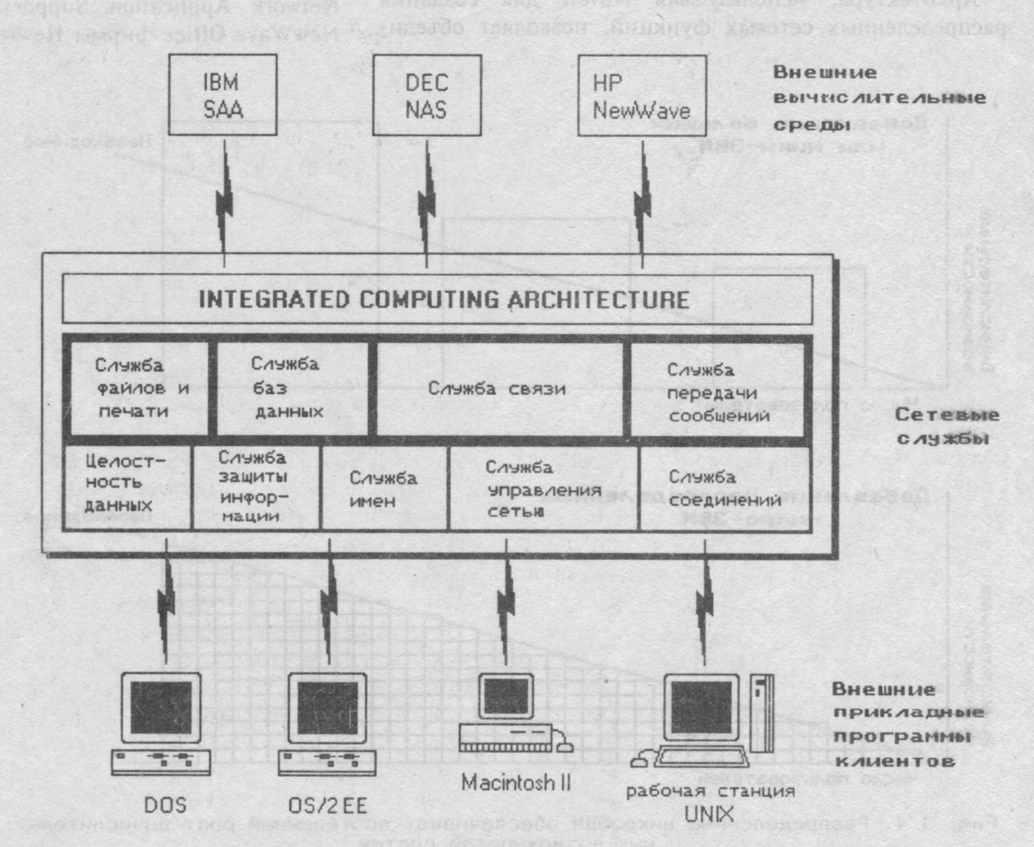


Рис. 1.6. Сетевые функции интегрируют прикладные программы сервера других поставщиков с прикладными программами клиентов

прикладных программ, совместно используемых на мини- и больших ЭВМ, в системы меньшего размера на основе распределенных микроЭВМ.

Специализированная распределенная сетевая операционная система NetWare содержит сервисные функции для выполнения прикладных программ у клиентов, а также функции выполнения прикладных программ на серверах.

Сетевые функции, определенные в архитектуре Novell Integrated Computing Architecture, могут выполняться как одним процессором сети NetWare, так и быть распределенными по локальной или региональной вычислительной сети. Каждая из этих функций может совместно использоваться другими прикладными программами или функциями.

На рис. 1.8 показаны распределенные между многими микроЭВМ функции, обеспечивающие объединение прикладных программ клиентов и сервера. При необходимости все функции NetWare могут распределяться одним микропроцессором, например, в случае удаленного офиса.

Novell Integrated Computing Architecture — это открытая архитектура, способная одновременно поддерживать различные стандартные протоколы. Рис. 1.9 иллюстрирует интеграцию прикладных программ, на-

Прикладной сервер

Адаптация

MVS,VM,VMS

OS/2EE

Рис. 1.7. Перенос прикладных программ с большой ЭВМ на прикладной сервер на базе микроЭВМ

писанных для разных протоколов, в рамках ядра протоколов NetWare.

Помимо поддержки различных протоколов, NICA также поддерживает "родные" файловые стандарты, что позволяет прикладным программам клиентов одновременно использовать один общий файл без необходимости разбиения диска на разделы. Поскольку архитектура Integrated Computing Architecture фирмы Novell является открытой и поддерживает стандарты сетевого управления, она позволяет осуществлять централизованное управление в распределенной среде.

NICA обеспечивает распределенные прикладные программы функциями поддержки целостности дан-

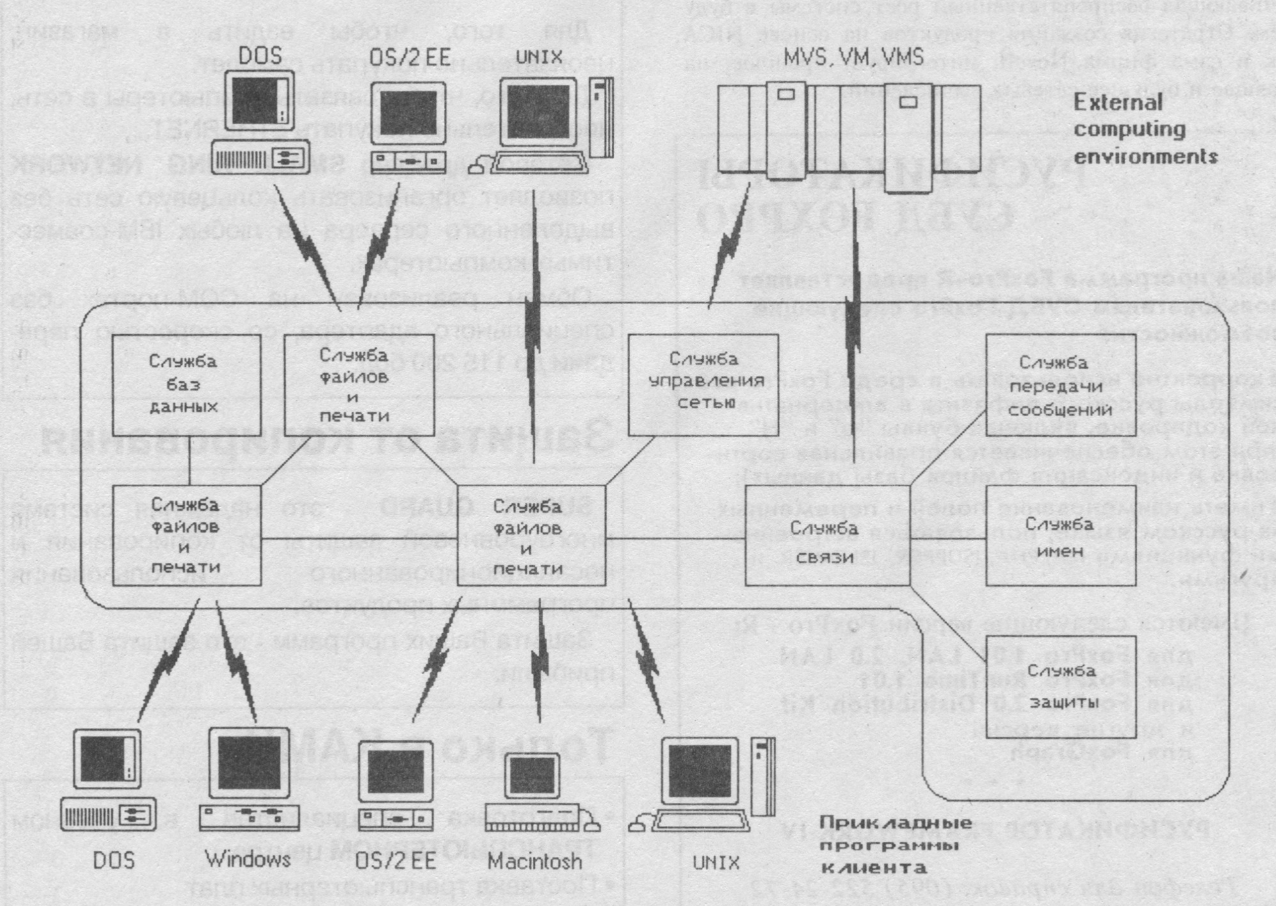


Рис. 1.8. NETWARE осуществляет распределение прикладных программ клиента и сервера

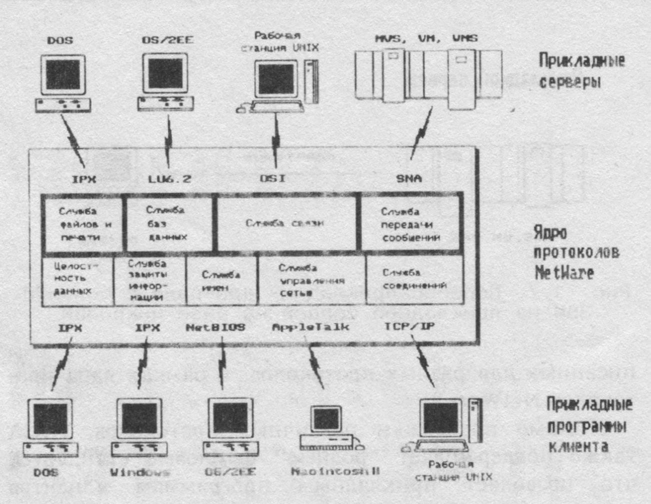


Рис. 1.9. Ядро протоколов NETWARE

ных, такими как зеркальное отображение дисков и серверов.

Используя архитектуру Novell Integrated Computing Architecture, менеджеры информационных систем корпораций могут чувствовать себя спокойно, поскольку это открытая архитектура, сохраняющая капитальные вложения в прикладные программы и обеспечивающая беспрепятственный рост системы в будущем. Стратегия создания продуктов на основе NICA, как и сама фирма Novell, интегрирует прошлое, настоящее и будущее сетевых вычислений.

СОВРЕМЕННЫЙ ПОТРЕБИТЕЛЬ: ЭВОЛЮЦИЯ ПОТРЕБНОСТЕЙ

Часто бывая в командировках в различных регионах страны, я много раз ловил себя на мысли, что отечественный рынок персональных компьютеров прямо на глазах претерпевает быстрые и серьезные изменения.

Народившийся совсем недавно класс покупателей персональных компьютеров в свои колыбельные времена норовил схватить слабыми ручонками то, до чего мог дотянуться. Цены на новоявленные диковинки — в то время — раритеты — были бы непостижимы уму простого западного обывателя, доведись ему с ними столкнуться. По законам экономики, которые, хотя и в рудиментарном виде, но все же присутствовали на тот момент в стране, спрос породил предложение, да еще какое! На волне хлынувшего в страну потока "персоналок" родились многочисленные СП и кооперативы, прибыли которых были поистине астрономическими.

А что же покупатель? Не имея выбора, он брал то, что предлагали, а предлагали, как мы сегодня знаем, отнюдь не шедевры западных технологий. Гиганты компьютерной промышленности, предлагающие действительно качественную технику, выходить на советский рынок не спешили (да и сегодня что-то не особенно торопятся), вполне обоснованно полагая, что рублевые прибыли, даже очень большие, но при этом не конвертируемые, едва ли можно считать прибылями. Поскольку цена была относительно стабильна на компьютеры вообще, а не на какие-то из них в особенности, закупались и ввозились в страну самые дешевые, в основном из Юго-Восточной Азии, откуда и пошло позднее презрительное название "желтые машины". О нуждах покупателя думать было некогда — нужно было "ковать железо, не отходя от кассы", — как говаривал один популярный персонаж.

Результат не замедлил сказаться. Тот, кто успел набрать компьютеров в числе первых, очень скоро столкнулся с проблемой, которая изначально была очевидной. Машины стали выходить из строя поодиночке и скопом, а обслуживать их было некому. Если машина была закуплена за рубежом (случай, кстати, почти небывалый), то для того, чтобы ее отремонтировать, необходимы были новые расходы — пересылка техники производителю и обратно порой обходится дороже, чем сам ремонт. Кооперативы, поставившие львиную долю машин, сорвав свой куш, стали вдруг закрываться один за другим. Головная боль счастливых владельцев "новейшей технологии" приобрела перманентный характер.

Сегодня осознание сложившейся ситуации пришло ко многим пользователям. Она начинает меняться и в большей степени походить на ту, что имеет место в странах с нормальной экономикой. Гиганты промышленности, особенно те из них, что располагают валютой, создают специальные подразделения и малые предприятия, единственной функцией которых является поиск иностранного поставщика техники и закупка оной для нужд учредителя. Такой подход, хотя и позволяет ускользнуть из загребущих рук разного рода посредников, тем не менее проблемы не решает. Сервис остается "основным вопросом современности". Прочие организации, не столь удачливые в плане создания продукции на пресловутом уровне мировых стандартов, а посему валюты не имеющие, мечутся по рынку из-за отсутствия информации, надеясь на то, что подвернется что-то действительно хорошее.

Смещение акцентов в подходе покупателя к выбору техники ста-

новится реальностью. В то время как бюджетные организации, сидящие на голодном пайке, еще цепляются за старые методы и норовят купить побольше и подешевле, на рынке появляется все больше заказчиков, вполне осознавших справедливость поговорки "я не настолько богат, чтобы покупать дешевые вещи". Немногочисленные пока компании, дающие на свою технику не только хорошую гарантию (полтора-два года), но и подкрепляющие ее быстрым и надежным сервисом, быстро находят своего клиента и имеют рынок с тенденцией к росту. Кроме того, все большее значение, похоже, начинает приобретать интеграция систем; машины в базовой конфигурации нужны все в меньшей и меньшей степени, растет потребность в комплексном решении проблем, стоящих перед заказчиком,— обеспечении предпродажных консультаций, сопровождения, поставки сетевого оборудования и прикладного программного обеспечения, самой разнообразной периферии.

Дифференциация заказчиков и их потребностей идет быстрыми темпами, значительно опережая предложение, существующее на рынке компьютерных услуг. На сегодняшний день в стране заметны лишь три крупных поставщика, предлагающих свою продукцию, несущих за нее всю полноту ответственности, обеспечивающих гарантию и сервис. Хотя эти три компании — ASI в Шуе, КОМПАН в Санкт-Петербурге и Summit Systems в Минске — очень разные, цели их схожи. ASI, созданная с участием немецких и тайваньских партнеров, выпускает продукцию на базе компонентов, ввозимых в основном из Азии, что позволяет снижать себестоимость при хорошем качестве. КОМПАН проводит похожую политику, но производит компьютеры собственной разработки. Summit Systems основана при участии американской компании Chips and Technologies, считающейся на американском рынке авангардом в области создания новейших компьютерных технологий. Продукция компании — это всегда новейшая техника, только появляющаяся на рынках Запада, а, значит, и цена на нее несколько выше обычной, что объясняется еще и наличием двухлетней гарантии и широкой сети дистрибьютеров по всей стране, обеспечивающих гарантийное и послегарантийное обслуживание. Благодаря всему этому данным фирмам удалось отыскать свои собственные ниши на советском рынке и избежать прямой конкуренции. При этом у них есть несколько общих черт — направленность на долгосрочную работу на советском рынке и ориентация на покупателя с действительно сформировавшимися запросами.

Появление таких компаний на рынке и сам факт их успешной работы в течение довольно длительного времени еще раз подтверждают изменения в структуре потребностей советского покупателя. Мы начинаем входить в новую стадию развития рынка компьютерных услуг — стадию ориентации на весь широкий спектр потребностей заказчиков, что само по себе означает куда большую эффективность "всеобщей компьютеризации", нежели мы могли наблюдать до сегодняшнего дня. Мы можем поздравить себя с вступлением в новый этап развития, дающий нам надежду, что и в нашей стране интересы и заботы заказчика возобладают над охотой за легкими прибылями.

А.Шарейко



НОВОСТИ

История о "вирусе для Хусейна"

Джон Маккормик сообщил Newsbytes, что U.S. News & World Report, Ted Koppel's Nightline, другие журналы и авторы многих книг — все попались на "первоапрельскую шутку", что американские военные якобы послали компьютерный вирус в компьютерную систему иракской системы ПВО во время операции "Буря в пустыне".

Сообщалось, что США, вероятно, поместили зараженную вирусом плату ПЗУ в принтер, который должен был быть провезен в Ирак в обход эмбарго ООН. Эта же история была напечатана в журнале "Infoworld" 1 апреля 1991 года на странице 39, где было указано, что это шутка. Если говорить о PC, то только принтер PostScript обычно может передать данные назад в компьютер, и даже если бы в таком принтере содержался вирус, маловероятно, чтобы это был выполнимый код, который мог бы быть перенесен на компьютер и активизирован. Однако использование вируса военными не так уж надумано. Несколько месяцев назад один профессор представил Национальному Институту по Стандартам и Технологии (NIST) доклад для семинара по компьютерной безопасности, в котором рассматривается использование вируса как оружия.

The Teleputing Hotline, January 27, 1991.

В тихом уголке выставки **Comnet** фирма **Quarterdeck Office Systems** продемонстрировала DESQview/X — гибрид своей популярной многозадачной оболочки DESQview под MS-DOS и X-Window под Unix.

Этот продукт может использоваться как под ДОСом, так и в Unix'е.

Фирма говорит, что под DESQview пользователь может получить реальную многозадачность и графический интерфейс, не покупая программ, специально изготовленных для многозадачной среды типа Windows.

Newsbytes News Network, January 29, 1992

Фирма **Intel** сообщила, что через ныне существующих в СНГ дистрибьюторов продано вчетверо больше микросхем, чем ожидалось ранее. Фирма будет увеличивать число своих агентов в СНГ и вложит в их обучение и воспитание в этом году около 2 миллионов долларов. Сейчас в московском офисе фирмы работают три человека, скоро их станет около 10.

Дистрибьюторами Intel являются: МП Квазар в Киеве, СП Диалог (Москва), Центр Техно (Москва), СП КАТ (Москва), МП Ком-Плюс (Зеленоград) и МП Невис в Санкт-Петербурге. В ближайшее время появятся еще пять дистрибьюторов. Большинство из них торгует микросхемами Intel только за валюту.

Проведя несколько семинаров по маркетингу для своих советских партнеров, фирма обнаружила, что лучше всего обучаются маркетингу отставные офицеры Советской Армии.

В этом году фирма переведет на русский язык и издаст достаточное количество справочных изданий о своей продукции.

Во всем мире в прошлом году фирма получила 7700 миллионов долларов прибыли.

Newsbytes News Network, January 30, 1992.

Фирма **IBM** выпустила пять новых мощных моделей серии RISC System/6000. Новое семейство Powerstation/Powerserver начинается с настольной машины, стоящей менее 7000 долларов, и заканчивается небольшой высокопроизводительной стойкой.

Самая дешевая на сегодняшний день RISC система — Powerstation/Powerserver 220 работает на процессоре с тактовой частотой 33 МГц и может использоваться для технических публикаций, разработки прикладных программ и работы на фондовой бирже. Машина имеет встроенные адаптеры Ethernet и SCSI, а также новый графический адаптер Power Gtl, поддерживающий работу с монохромным, черно-белым и цветным мониторами.

Минимальная версия рабочей станции с жестким диском емкостью 400 Мбайт и 16 Мбайт ОЗУ стоит 9995 долларов.

Разница между Powerstation и Powerserver состоит в том, что первая поставляется с графическим адаптером и дисплеем, тогда как последняя — без всего этого, но с четвертьдюймовым стримером.

Более мощные машины моделей 340 и 350 также выполнены как настольные устройства. Модель 340 имеет тактовую частоту 33 МГц и цену 18 895 долларов, тогда как модель 350 (43 МГц) стоит 26 895 долларов в конфигурации с 16 Мбайт ОЗУ и 160 Мбайт винчестером.

Модель 530H работает на 20 процентов быстрее, чем существующая модель 520, имеет 25 МГц процессор, 16 Мбайт ОЗУ и 400 Мбайт винчестер. Цена — 28 110 долларов.

Но самым мощным из последних шедевров фирмы является Powerstation/Powerserver 560, сделанный на 50 МГц RISC-процессоре. Эта новая настольная модель, по заявлению фирмы, обеспечивает производительность 89,3 SPECmarks, или 30,4 MFLOPS. Стоит все это удовольствие 64 110 долларов в конфигурации рабочей станции или 62 240 долларов для сервера.

На программном фронте у фирмы тоже есть новости. Недавно она объявила о выпуске версии операционной системы Unix — AIX/6000 3.2, которая является первой операционной системой, выпущенной в полном соответствии с утвержденной Фондом Открытого Программного Обеспечения Спецификацией Прикладных Программ (Application Environment Specification).

Среди других программных продуктов фирмы — CASE оболочка AIX Software Development Environment и средства создания онлайновых библиотек данных, доступных для всех пользователей внутренней компьютерной сети предприятия.

Newsbytes News Network, January 27, 1992.

Американская пресса, компьютерные специалисты, сотрудники публичных библиотек протестуют против решения Комиссии по ценным бумагам о создании электронной системы сбора, обработки и хранения информации. Эта система выполняется согласно новой концепции американского правительства о приватизации средств распространения правительственной информации.

Представители ассоциации налогоплательщиков считают, что при распространении правительственной информации коммерческой фирмой, занимающейся коммерческой продажей подобной информации, могут возникнуть разногласия интересов, которые будут вести к ущемлению прав налогоплательщиков.

Newsbytes News Network, January 27, 1992.

Американская компания DFM изготовила переносной компьютер Travelite, который может по радио присоединяться к внутрифирменной сети Novell. Размеры этого компьютера — 21.3 x 31.0 x 4.0 см, а весит он всего 3 кг. Travelite позволяет отходить от базовой станции сети до полукилометра, оставаясь при этом подключенным к ней. Скорость передачи информации — 24 или 121 килобайт в секунду.

Travelite выполнен на базе микропроцессора 80286, работающего на тактовой частоте 12 МГц. В стандартную конфигурацию входят: чувствительный к прикосновению руки экран для ввода информации с разрешением 1000 x 1000 точек, жидкокристаллический EGA-монитор с подсветкой, 2 Мбайта ОЗУ, 512 Кбайт ПЗУ с программами вместо привода A:, винчестер на 20 Мбайт, отдельная клавиатура, батарея и блок питания. Все это стоит 3466 долларов. В качестве дополнительных аксессуаров могут быть куплены диск до 80 Мбайт, 16 МГц процессор, сопроцессор, встроенный факс-модем, цифровая плата для передачи голоса, до 8 Мбайт ОЗУ, микрофон, последовательный порт, считыватель штрихового кода и отдельный монитор.

Newsbytes News Network, January 27, 1992.

Adobe объявила о выпуске Adobe Streamline 3.0 для Microsoft Windows — программы преобразования цветных и черно-белых точечных изображений в формат Postscript.

Новые свойства пакета включают в себя средства обработки изображений, поддержку большего числа цветов, более приятный интерфейс, экспорт большинства популярных графических форматов и возможность одновременно работать с Adobe Illustrator.

Для преобразования изображений применены три различных метода. Контрастная конверсия, основанная на разнице цветов и их интенсивностей, применяемая для контрастных изображений типа фотографий.

Второй алгоритм конверсии — threshold — разбивает весь спектр на равное число частей. Фирма заявляет, что это бывает полезно для изображений, содержащих много разных цветов. Третий тип — posterization — позволяет переходить от 256 цветов к 16 или 2 градациям серого.

Пакет имеет развитые средства обработки изображений до и после преобразования. Есть возможность подстройки цветов до начала конверсии, что удобно для ретуширования фотографий — например, можно взять цвет щек и использовать именно его для ликвидации теней под глазами.

После обработки можно заняться сглаживанием кривых, поврежденных при переходе от точечного формата.

Имеется также возможность сканирования документов и форм, причем программа может работать и тогда, когда документ был отсканирован с небольшими отклонениями от вертикали или горизонтали.

Newsbytes News Network, January 27, 1992.

IBM выпустила новые продукты для локальных сетей. Для использования в качестве моста или рутера предназначен IBM 6611 model 140/170, который поддерживает большинство известных сетевых протоколов — TCP/IP, SNA, NetBIOS, AppleTalk, IPX, DECnet и XNS. Компания планирует в будущем расширять этот

список, покупая продукты для межсетевого взаимодействия других фирм. IBM 6611 будет продаваться за 10 000 долларов начиная с июня этого года.

Третий новый продукт — это рабочая станция менеджера сети AIX NetView/6000. Она выполнена на базе RISC SYSTEM/6000 и включает в себя протоколы TCP/IP и SMNP.

Newsbytes News Network, January 27, 1992

Фирма **Microsoft** впервые в своей истории решила дать рекламу по телевидению. Для благой цели — рекламы Windows 3.0 — начиная с марта будет потрачено около 8 миллионов долларов. До настоящего времени было продано около 9 миллионов копий этой программной оболочки. Реклама будет предназначена для тех, кто не читает компьютерных журналов, но использует машины в работе. Компания идет на такие затраты даже несмотря на то, что компьютерный рынок США сильно страдает из-за экономического спада, поразившего страну.

Newsbytes News Network, January 27, 1992

Texas Instruments увольняет еще 1000 человек в отделах полупроводников и информационных технологий.

За прошлый год фирма уволила 5 тысяч человек, оставив в штате всего около 62 тысяч сотрудников.

Одновременно фирма объявила, что в 1991 году понесла 409 миллионов долларов убытков. Причиной всего называется небывалый спад производства, продолжающийся в США уже около года.

Newsbytes News Network, January 27, 1992

Советская армия может с небольшим "боем" отдать часть диапазона частот в районе 900 МГц, которая во всем мире используется для сотовых сетей связи, а в СНГ — для военной аэронавигации. Предположительно, военные отдадут те части спектра, которые были ими взяты, но на самом деле не использовались. Как стало известно агентству Newsbytes, кроме организаций, имеющих целью создание общественных сотовых телефонных сетей, борьбу за 900 МГц ведут также милиция и служба государственной безопасности, которые хотят улучшить свои собственные коммуникационные системы.

Newsbytes News Network, January 27, 1992

Адреса книжных магазинов — опорных пунктов агенства "Компьютерпресс" по распространению журнала "Компьютерпресс":

1. 117334 Москва, Ленинский проспект, 40, "Техническая книга", тел. 137-60-19
2. 101000 Москва, Мясницкая, 6, "Книжный мир"
3. Москва, Чернышевского, 44, М-н №206
4. 630076 Новосибирск, Красный проспект, 60, "Техническая книга", отдел "Книга почтой"
5. 620151 Свердловск, К.Либкнехта, 16, "Техническая книга"
6. 191186 Санкт-Петербург, Невский проспект, 28, "Дом книги"
7. 220005 Минск, Ленинский проспект, 48, "Техническая книга"

Беспроводная передача данных

В 1992 году начнут эксплуатироваться различные методы беспроводной передачи данных на различных частотах. BellSouth приобрела половину радиосети передачи данных компании Ram Mobile Data за 300 миллионов долларов, а IBM и Motorola стоят за конкурирующей сетью Ardis. Есть еще компания Cellular Data Inc., которая собирается предложить пакетную передачу данных по обычным сотовым телефонным сетям, возможно, вместе с фирмами-операторами сетей. Прибавьте сюда Fleet Call и другие системы предачи данных ESMR, и получите огромные суммы капиталов в погоне за рынком, существование которого никто не может доказать.

The Teleputing Hotline, January 27, 1991.

Infonet и Sears связываются через EDI

Infonet, пакетная сеть, принадлежащая MCI, и консорциум международных телефонных систем подписали соглашение с Sears Communications, подразделением громадного концерна, о союзе в области электронного ведения платежных документов (EDI). EDI из Infonet охватывает около 118 стран, а Sears заправляет EDI для корпораций. По условиям соглашения Infonet будет подтверждать принимаемые сообщения из Европы и азиатско- тихоокеанского региона, а SCC будет подтверждать доставку исходящих сообщений. SCC будет также обрабатывать запросы от всех своих североамериканских клиентов, которые хотят соединить своих поставщиков и дистрибуторов на других континентах с помощью EDI со службой сети Infonet Notice EDI.

The Teleputing Hotline, January 27, 1991.

В Гонконге **Citibank Plaza** в марте откроет первое "умное" здание в городе. Встроенная телефонная система обеспечивает все этажи услугами по передаче голосовых и цифровых данных, предоставляет доступ к данным, передаваемым по спутниковой и кабельной связи, а также к информационным службам. Кроме этого, фирма CSSL выпустила два новых программных продукта, обеспечивающие проведение сделок. Фирма заявляет, что они лучшие в мире. Financial Information Systems Toolkit (FIST) и FS-Dealer, разработанный английской компанией Kapiti Systems, продаются вместе с глобальной банковской системой Equation этой же компании. Hong Kong Telecom выпустил серию телефонных карточек, изображающих символы традиционного китайского календаря. Карточки, которые могут использоваться для телефонных звонков, становятся, как почтовые марки, предметом коллекционирования. Каждая из этих карточек выпускается ограниченным тиражом (100 000), первая из них имеет номинал 50 гонконгских (6,5 американских) долларов.

The Teleputing Hotline, January 20, 1991.

К.Чащин



Кроме того, Движение — один из девизов фирмы SHARP.
Совсем недавно в жизнь общества вошли персональные компьютеры. Их появление произвело революцию в жизни человечества. И вот мы уже работаем с компьютерами-блокнотами. Их появление стало возможным благодаря удивительному прогрессу в различных областях электроники и технологии. Фирма SHARP, после тщательных исследовательских работ, объединила превосходную электронику с лучшим в мире жидкокристаллическим экраном, получив компактный и мощный компьютер, который не заставляет Вас сидеть в офисе. В бизнесе важно оказаться вовремя в нужном месте. Вы всегда будете по всеоружии появляться в нужном месте, опережая конкурентов. В конечном счете, Вы сможете делать свое дело лучше и быстрее. SHARP позаботилась об этом.
Если Вы считаете, что главное в бизнесе — бизнес, а не отсиживание времени в солидном офисе за огромным столом мореного дуба, то этот компьютер — для Вас.

SHARP


МАСТАК
ИСМ-1200